CONTIENE:

AMIGA Disk Magazine

Pixmate

Form Ilbm

SMAU 88

Kindwords

Spedizione in abbonamento postale Gruppo III/70 Tassa percue (Tassa riscossa) Milano Ferr. Corr.

EDITORIALE

Questo terzo è il primo numero della rivista ad uscire dopo lo SMAU e pertanto, ci troverete un breve resoconto, dal punto di vista dell'utenza Amiga, di quanto di buono, non moltissimo per il vero, si sia potuto vedere in questa grande rassegna dell'informatica nazionale. L'attuale situazione Amiga non è delle più allegre, ma per approfondire ciò vi rimando all'articolo: per ora veniamo ad altro.
Oltre agli articoli oramai familiari sul Basic, Assembly, C e CLI, molte altre sono le chicche contenute in questo numero. I nostri amici Michelon e Manzo per esempio, quei due "pazzi" che si divertono a utilizzare il computer per risolvere equazioni di ogni grado e genere, questo mese propongono un programma che vi permetterà di "creare" un piccolo universo in miniatura, dotato di soli, pianeti e satelliti: esattamente come quello in cui svolazziamo da qualche miliardo di anni, o quasi.
La redazione rende noto che non rifonderà in nessun caso il costo di eventuali cure psichiatriche. Per chi invece utilizza il proprio computer per fini meno appariscenti, come per scrivere, disegnare ed altro, presentiamo il primo word processor interamente in italiano e, cosa non indifferente, assolutamente in PAL: Kind Words, prodotto e distribuito dalla C.T.O. di Bologna. Per chi è invece un appassionato della programmazione gli articoli da elencare sarebbero troppi e pertanto preferisco lasciare a voi il compito di scoprirli tra le pagine della rivista.
Nel salutarvi, vi annuncio sbalorditivi, pirotecnici, strabilianti, megagalattici... (al momento non mi vengono in mentre altri aggettivi, per cui la smetto qui) rinnovamenti già dal prossimo numero. Tanto per cominciare ci sarà..., poi non ci sarà..., e quindi potrete... Ci ho ripensato (risatina sadica; la prossima volta ve la campiono sul dischetto per farvela ascoltare), non ve lo dico: lo scoprirete da voi il prossimo mese. Arrivederci.

La Redazione

SOMMARIO

Foto di copertina realizzata dallo studio artistico della:
Graphic & Comp. di Gorizia

Anno I numero 3 Novembre 1988

DIRETTORE RESPONSABILE
Giampietro Zanga

REDAZIONE
Graphic & Comp. Gorizia

COORDINAMENTO REDAZIONALE
Simone Concina

ART DIRECTOR
Gianni Marega

COLLABORATORI
Roberto Beccia, Primoz Beltram, Tomi Beltram, Fabio Biancotto, Giorgio Dose, Mr. Lambda, Massimo Lavarin, Furio Lusnig Luigi Manzo, Giovanni Michelon, Emilio Orione, Alessandro Prandi, Giacomo Pueroni, Paolo Russo.

GRAFICA, IMPAGINAZIONE, COPERTINA
Graphic & Comp.

REVISIONE GRAFICA E IMPAGINAZIONE
Gianni De Tomasi

DIVISIONE PUBBLICITÀ
Via Pola 9 - 20124 MILANO - Tel. 69.481
Telex 316213 REINAI - 333436 GEJ - ITI
OVERSEAS DEPARTMENT: Tel. 02/6948492
PUBBLICITÀ PER ROMA-LAZIO E CENTRO SUD
Via Lago diTana, 16 - 00199 Roma
Tel. (06) 8380547 - Telefax (06) 8948438

FOTOCOMPOSIZIONE
FOTOFORMA - Via del Molino a Vento, 72
34137TRIESTE

STAMPA
Grafika '78 - Pioltello (MI)

DISTRIBUZIONE
Sodip- Via Zuratti, 25 - 20125 MILANO
Spedizione in abbonamento postale Gruppo III/70

UFFICIO ABBONAMENTI
Tel. (02) 8122527-6187376
Prezzo della rivista L. 14.000 (Frs. 21.00)
Numero arretrato L. 28.000
Abbonamantoannuo L. 92.500
per l'Estero L. 185.000
I versamenti vanno indirizzatia:
GruppoEditoriale Jackson
Via Rosellini, 12 - 20124 Milano
mediante emissione di assegnobancario, vaglia o utilizzando il C/C/ postale numero 11666203
Peri cambi di indirizzo, indicare, oltre alnuovo, anche l'indirizzo precedente.

PUBLISHER
Filippo Canavese

DIREZIONE
REDAZIONE, AMMINISTRAZIONE
Via Rosellini, 12 - 20124 Milano
Tel (02) 6680000-66800181 - Telex 333436 GEJIT I
SEDE LEGALE
Via Pietro Mascagni, 14 - 20122 Milano
Il Gruppo Editoriale Jackson è iscritto nel Registro nazionale della Stampa al n. 117 vol. 2 - foglio 129 in data 17/8/1982

Autorizzazione del Tribunale di Milanon. 102 del 22/2/1988

AMIGA NEWS

Accordo C.T.O. - Infogrames

Nel luglio 1988, la società francese INFOGRAMES, leader nel settore dei video-game per personal computer, e la società C.T.O. di Bologna, hanno firmato degli accordi per la distribuzione esclusiva dei prodotti Infogrames in Italia.

Una gamma di software prestigiosi, tra i quali le ultime novità ancora inedite in Italia, sarà presentata allo SMAU 1988 allo stand della COMMODORE ITALIA.

Per il momento, la distribuzione essclusiva riguarderà la gamma Commodore (C-64 e Amiga) e i compatibili PC; i prodotti presentati saranno i seguenti: CAPTAIN BLOOD, MACADAM BUMPER, AQUAMAN VERSUS THE YELLOW SHADOW, BOBO, I PASSEGGERI DEL VENTO, BUBBLE GHOST, WARLOCK'S QUEST, SPIDERTRONIC, CAD 3D, LAST MISSION, LIVINGSTONE.

Partner d'eccezione per prodotti eccezionali

Il gruppo editoriale INFOGRAMES è nato attraverso la fusione di tre società leader nel settore europeo dei videogame:

INFOGRAMES,
ERE INFORMATIQUE,
COBRA SOFT.

Il gruppo Infogrames è già presente in 15 paesi e realizza più del 30% della sua cifra d'affari con l'export. Tra le caratteristiche principali della sua attività ricordiamo una costante ricerca nella creatività dei video giochi, specie tramite l'abbinamento col mondo del fumetto, del cinema, dei cartoni animati e dello sport.

Un centinaio di collaboratori compongono la sua struttura per un fatturato che nel 1987 ha raggiunto la cifra di circa 15 milioni di dollari.

Il gruppo C.T.O. è una società estremamente dinamica e in continua crescita con sede a Bologna ed è rivolta al mercato del software attraverso due comparti principali:

- I prodotti professionali del mercato MS/DOS per quanto concerne la produttività individuale,

- i prodotti "consumer".

La base dell'attività del C.T.O. consiste nella ricerca, nello studio e nella commercializzazione di prodotti per l'informatica, direttamente importati e selezionati fra quelli delle migliori case produttrici mondiali.

I prodotti INFOGRAMES vanno ad aggiungersi ad altri importanti marchi esclusivi per l'Italia, tra i quali citiamo:

— LOGISTIX,
— PORTEX,
— PRECISION SOFTWARE LTD,
— LIFETREE SOFTWARE LTD,
— ELECTRONIC ARTS,
— PROGRESSIVE PERIPHERALS & SOFTWARE.

Nonché l'affermata linea COMMODORE SOFTWARE by C.T.O.

La flessibilità della struttura del C.T.O. si traduce in scelte di mercato che consentano una risposta efficace e funzionale alla domanda del settore informatico.

La rete di vendita del C.T.O. copre la totalità del mercato nazionale.

Per ulteriori informazioni, contattare:

INFOGRAMES : Sabine Robert (Public Relations)
84 rue du 1er mars 1943
69628 Villeurbanne Cedex
tel: 00 33 78 03 18 46
telex: 305 551
fax: 00 33 78 03 18 40

C.T.O. : Maria Rosa Piovani - Agenzia Technostudi
via Mascarella, n.57
40126 Bologna
tel: (051) 241 436

Sussidi Hard

Mentre la Commodore sta espandendo la sua linea di prodotti Amiga, la Phoenix Electronics amplia la linea delle proprie macchine individuali. Tra gli ultimi progetti presentati sono degni di nota gli hard drive da 20 e 40 MB. Questi hard drive sono utilizzabili con l'Amiga 500 (modello PHD 500) e con l'Amiga 1000 (modello PHD 1000). I prodotti sono coperti da garanzia per la sostituzione di parti, compreso il lavoro, per la durata di un anno. Entrambi i drive di colore beige sono in possesso di un interfaccia SCSI e vengono forniti con dei programmi dimostrativi e con del software di pubblico dominio. Sono disponibili anche dei modelli a Direct Memory Access. Il drive va collegato alla porta d'espansione posta sulla sinistra e permette comunque il collegamento per ulteriori espansioni esterne.

Il PHD 500/20 viene venduto al prezzo di $949 e il PHD 500/40 a $1.399. Il PHD 1000/20 viene venduto al prezzo di $969 e il PHD 1000/40 a $1.429.

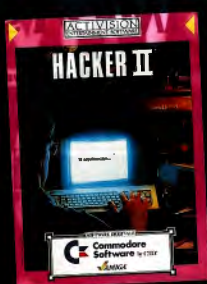
Per ulteriori chiarimenti contattate la Phoenix Electronics Inc., P.O. Box 156, Clay Center, KS 67432, USA.

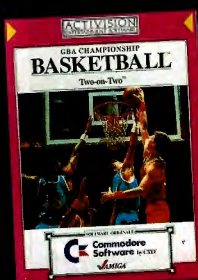
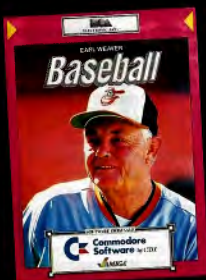
Animation Stand

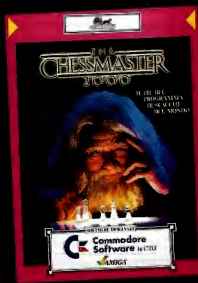
Il programma permette di produrre delle immagini scegliendo tra 4096 colori. È completamente compatibile con altri prodotti Animation ed è perciò possibile introdurre qualsiasi disegno con le varie risoluzioni e modi grafici.

Una speciale tecnica di interpolazione è utilizzata per debellare il problema legato ai pixel durante la fase di zoom all'interno di una figura, in modo da non perdere l'integrità dei dettagli del disegno.

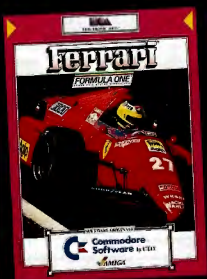
Il programma viene venduto al prezzo di 35 sterline ed è disponibile presso l'Amiga Centre Scotland, 4 Hart Street Lane, Edinburgh EH1 3RN, Inghilterra.

Tutti i programmi originali della Collana Commodore Software by C.T.O. li troverete presso i Commodore Point.

Hacker II
Lit. 29.500

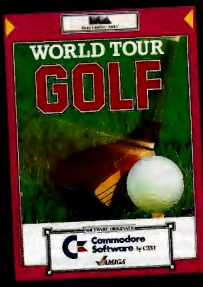
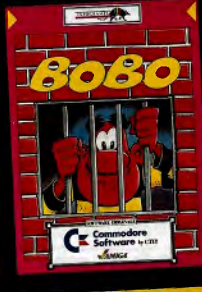
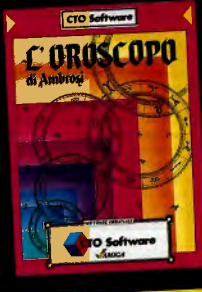
Shanghai
Lit. 29.500

GBA Championship Basketball
Lit. 29.500

Earl Weaver Baseball
Lit. 29.500

Test Drive
Lit. 33.000

ChessMaster 2000
Lit. 29.500

Skyfox
Lit. 29.500

Marble Madness
Lit. 29.500

Ferrari Formula One
Lit. 38.000

Return to Atlantis
Lit. 40.000

Chiedi ogni mese al tuo Commodore Point le novità

AMIGA TRICKS

Auto notepad come promemoria

Se assomigliate al sottoscritto allora avete sempre la necessità che qualcuno vi ricordi i vari compiti che dovete svolgere durante l'arco della giornata. Dal momento che questo fatto risulta essere un compito tutt'altro che semplice eccovi un modo molto informale per mantenere la traccia delle varie faccende che dovete sbrigare.

Dal momento che la nuova versione del Notepad può essere aperta per mezzo del CLI, è possibile includervi una sequenza di startup. Non dovrete far altro che aggiungere la seguente linea:

NOTEPAD filename

prima del comando LOADWB nella sequenza di startup. In questo modo ogni volta che avvierete il vostro Amiga, il notepad con le vostre note verrà visualizzato, potrete così aggiungervi dei nuovi item o qualsiasi altra cosa che riteniate necessario, e quando uscirete dal Notepad, il Workbench verrà caricato in modo automatico.

Migliorate i colori nelle stampe

Nell'utilizzo delle stampanti a volte la qualità della definizione dei colori sulla carta non risulta come la si vorrebbe, fatto questo che farebbe spazientire anche le persone più tolleranti. Ci sono però alcuni accorgimenti che possono migliorare in maniera notevole la stampa dei vari colori.

Primo, se la vostra stampante utilizza qualsiasi tipo di carta in commercio, effettuate delle prove tra le varie carte di tipo commerciale, in modo da trovare quella che dà una miglior resa del colore dal momento che non tutti i tipi di carta reagiscono alla stessa maniera.

Secondo, spruzzate le vostre stampe con dello spray Krylon. Otterrete come risultato una miglior saturazione del colore, un maggior contrasto e un incredibile senso di profondità.

Flickering dello schermo: soluzioni!

Ci sono state molte lamentele per quanto riguarda il flickering dello schermo quando si utilizza l'Amiga nel modo interlacciato. Se non avete l'intenzione di spendere del denaro per l'acquisto di un monitor ad alta persistenza, esistono alcune soluzioni molto meno costose.

Primo, provate a posizionare un sottile foglio di plexigas color grigio fumo davanti allo schermo del monitor.

Secondo, provate ad acquistare un paio di occhiali da sole ma che non siano troppo scuri e che siano allo stesso tempo degli occhiali polarizzati. Entrambi questi metodi riducono notevolmente il flickering senza per questo alterare in modo significativo i colori dello schermo.

Flickering e Preference

Nel provare la prima volta una copia del Workbench 1.2, la sensazione è identica a quella provata quando si utilizza una nuova macchina. Una delle configurazioni che si può verificare per prima è quella di operare con il modo interlacciato, ma si rischia di essere travolti dal flickering.

Dopo molte prove ed esperimenti, abbiamo trovato che la scelta di una buona combinazione di colori del Preference riduce al minimo il flickering, o perlomeno il disturbo diventa minimo. Provate perciò a settare i registri da uno a quattro nel modo seguente:

1	2	3	4
R G B	R G B	R G B	R G B
0 4 6	0 0 0	0 0 9	15 0 0

Per prima cosa settate il colore a zero, poi premete il pulsante del mouse il numero di volte che vi viene indicato sulla parte destra della barra del colore. Il registro 1 è il registro che appare all'estrema sinistra sullo schermo delle Preference.

Osservando il Kickstart

Il dischetto del Kickstart contiene nel suo interno il sistema operativo Basic dell'Amiga. Questi viene caricato all'interno dell'Amiga nel punto più alto dello spazio per gli indirizzi, dalla locazione di memoria 16.580.608 alla 16.777.216 (avete letto giusto la cifra menzionata non è un errore di stampa, sono esattamente 16 milioni!).

Curiosando nel programma, vengono alla luce alcuni fatti interessanti. Provate il seguente. Caricate l'Amiga Basic e digitate il seguente breve programma:

```
start = 16653596: quit = 16653672:
CLS:
  WIDTH 67: PRINT
for j = startto quit
  PRINT CHR$(PEEK(j));
  NEXT j
```

Gli indirizzi di start e di quit menzionati sono per il Kickstart 1.1.

Sostituiteli nel seguente modo se intendete operare con il Kickstart 1.2: start = 16649670 e quit = 16649747.

Basic Clear

Se utilizzate il comando CLEAR per predisporre dello spazio di lavoro per i vostri programmi in Amiga Basic (es. CLEAR ,150000), e poi cercate di effettuare per due volte in una riga il run del vostro programma. Otterrete un messaggio, a prima vista inspiegabile, del tipo OUT OF MEMORY, e questo anche se non avete modificato il programma tra l'esecuzione dei due run. Ci si chiederà perciò se c'è memoria per eseguire il run del programma la prima volta, come mai non ce n'è per la seconda esecuzione?

La causa è da ricercarsi nel fatto che il Basic dell'Amiga riserva una prima volta un'area di 150.000 byte mentre tenta di eseguire il secondo CLEAR, 150000 (mantiene il testo del vostro programma nella prima area finché non può spostarlo nella seconda). Si finisce con aver bisogno di 300.000 byte per eseguire il secondo run.

La soluzione potrebbe essere la seguente: sostituire il comando CLEAR,150000 con:

```
CLEAR ,X
CLEAR ,150000
```

Dove 'X' è grande abbastanza per contenere il testo del vostro programma. In questo modo Il Basic dell'Amiga necessita di 150.000 + X byte per effettuare il run la seconda volta.

Fate attenzione che 'X' sia abbastanza grande o vi troverete nuovamente davanti al messaggio di OUT OF MEMORY che richiederà un nuovo caricamento in memoria del sistema. È possibile rintracciare l'ampiezza necessaria della 'X' caricando semplicemente il vostro programma, poi eseguirlo nel modo immediato e digitare infine:

```
CLEAR ,Y
PRINT Y-FRE(0)
```

dove 'Y' è noto essere delle dimensioni del vostro programma. L'ampiezza del vostro programma verrà così stampata, ricordatevi di aggiungere a questa una cifra che possa contenere ragionevolmente le vostre future revisioni.

Cancellare Directory

Questo è un metodo semplice e rapido per cancellare delle directory non desiderate. Digitate il seguente comando dal CLI finché siete all'infuori della directory che intendete eliminare:

```
DELETE pathname ALL
```

dove il pathname è il nome completo della directory che intendete cancellare. Questo comando cancellerà tutti i file e le subdirectory all'interno della directory menzionata e in seguito cancellerà la stessa directory.

Ad esempio, supponete di voler eliminare RAM:C. Dopo esservi spostati dalla RAM:C (utilizzando CD), digitate:

```
DELETE RAM:C ALL
```

Tutti i file contenuti nella RAM:C saranno cancellati, ed in ultimo la directory verrà rimossa. Questo tipo di comando opera altrettanto bene con i file del dischetto. Se digitate:

```
DELETE DF0:C ALL
```

con il disco Workbench posto nel drive interno del vostro Amiga, la directory C verrà cancellata. Questo comando si rivela molto utile per l'editing dei dischetti, è molto veloce e non è reversibile, perciò siate certi delle vostre intenzioni prima di premere il tasto Return altrimenti rischiate di perdere dei file preziosi.

Velocizzare l'emulatore

Se avete provato invano a migliorare la velocità dell'Emulator Transformer, vi sarà possibile trovare una risposta nel manuale IBM DOS Reference (versione 3.10). Molti programmi in possesso di notevoli capacità operative richiedono il cambiamento del numero dei Buffer e dei File, ma modificare il Device per utilizzare ANSI.SYS è possibile solo se la tastiera è di modello non standardizzato, ed è questo il caso dell'Amiga.

Vi proponiamo l'esatta serie di comandi da utilizzarsi:

```
COPY CON CONFIG.SYS
BUFFERS = 20
DEVICE = ANSI.SYS
FILES = 20
(F6) or (CTRL-Z)
```

Il manuale afferma che dopo la prima linea del comando, non ci sarà il prompt usuale; esso riapparirà dopo l'ultima linea con il numero dei file copiati. Quest'operazione migliorerà in modo notevole la velocità facendovi risparmiare del tempo.

Comandi abbreviati

Vi proponiamo una scorciatoia che riteniamo utile quando si utilizza il CLI. Ci si posizioni nella directory in C e si copi una serie di comandi, fornendo loro dei nomi abbreviati. Ad esempio:

```
COPY C:DIR TO C:D
COPY C:ENDCLI TO C:END
COPY C:DELETE TO C:DEL
```

Ora se volete sbarazzarvi di alcuni file che ritenete non possano esservi più di alcuna utilità, dovrete semplicemente digitare il comando DEL al poste del comando DELETE. Ciascun comando che copiate occuperà lo spazio di un blocco. Potete anche effettuare il RENAME dei comandi, ma i programmi potrebbero avere bisogno del loro vecchio nome, perciò fate attenzione.

Basic Baud

Nello scrivere dei programmi di bulletin board con l'Amiga Basic, ci si può trovare nella necessità di definire un modo per settare la velocità baud. Le seguenti POKE risolveranno per voi questo problema:

```
POKE 14676018, 12000 =  300 baud
POKE 14676018,  3000 = 1200 baud
POKE 14676018,  1500 = 2400 baud
```

Ci sono naturalmente molti altri valori che potrete utilizzare, ma quelli che vi abbiamo proposto sono i più comuni.

Save di Icone aggiornate

Questo è un metodo relativamente semplice per mantenere la stessa icona dopo averne creata una nuova. Normalmente, il Basic cancella la vecchia icona quando riregistrate il programma, ma questo fatto può essere evitato in modo abbastanza semplice e permanente, basterà che digitate:

```
PROTECT programname.info R
```

Questo attiverà la protezione per il file info, dove è immagazzinata l'icona, in questo modo il file info non potrà essere cancellato finché la protezione non venga rimossa per mezzo del seguente comando:

```
PROTECT programname.info RWD
```

Caleidoscopio

Se desiderate stampare alcune schermate generate con il programma Electronic Arts' Kaleidoscope, vi accorgerete immediatamente e con rammarico che ciò non vi sarà possibile. Il motivo è relativamente semplice, non ci sono Preference sul dischetto.

Per aggirare l'ostacolo non dovrete far altro che copiare il programma su un dischetto vuoto, scegliere le Preference desiderate e copiarle sullo stesso dischetto sul quale avete precedentemente copiato il programma. In questo modo avrete risolto il problema nel quale eravate incappati.

Deluxe Paint e Aegis Images

I programmi Deluxe Paint e Aegis Images si completano uno con l'altro in modo perfetto ed è possibile così utilizzare le diverse configurazioni offerte dai due programmi. Nel creare dei disegni l'artista può passare da un programma all'altro traendo in questo modo il massimo vantaggio dalle opzioni offerte da entrambi i programmi, realizzando così dei disegni che si avvalgono di un numero maggiore di peculiarità.

Utilizzate un programma per creare una figura, registratela su dischetto, passate ora all'altro programma, ora ricaricate in memoria la figura e continuate la vostra opera. Utilizzando il programma Deluxe Paint non incontrerete alcun problema nel caricare in memoria disegni formati con l'Aegis Images (naturalmente i due disegni dovranno essere nella medesima risoluzione).

Images invece, è programmato in modo da appendere in maniera automatica un suffisso del tipo '.pic' o '.hpic' ai nomi dei disegni e riconosce e carica in memoria solo dei file in possesso di uno di questi due suffissi. Per essere in grado di utilizzare il programma Aegis Images con dei disegni iniziati con il Deluxe Paint, dovrete registrare i disegni formati con il Deluxe Paint con il nome terminante in '.pic' (per il modo 320×200) o con '.hpic' (per il modo 640×200) in questo modo i disegni saranno accessibili anche dal programma Images.

Operation neptune

Oceano Pacifico, settore Nord vicino alle isole Marshall, ore 6.15 pm.

A bordo di un Hercules C-130-S, Aquaman sta per paracadutarsi sull'oceano. Dopo esser ammarato dovrà affrontare un duello mortale guidando il suo scooter d'acqua contro un uomo inviato da Yellow Shadow, un personaggio crudele che desidera diventare il padrone della terra.

Il duello è appena iniziato e Aquaman durante la sua discesa nell'oceano dovrà affrontare gli uomini rana, le piovre provviste di tentacoli velenosi, gli squali e altri mostri.

Il gioco è composto da sette diverse parti:

— un duello su scooters,

— una caccia a bordo di un batiscafo,

— spostarsi all'interno di zone di mare minate,

— rintracciare sulle mappe i nemici,

— lottare con nemici di diversa natura: uomini rana, squali, piovre, ecc.,

— battaglie con scooter sottomarini forniti di torpedo nucleari,

— ripulire dalle mine il mare circostante le proprie basi e danneggiare o distruggere le basi nemiche.

Il game appartiene alla nuova generazione di software per microcomputer dove i giochi sono collegati gli uni agli altri come in un film.

La grafica viene visualizzata sull'intero schermo e il sistema di lotta tra i personaggi è molto realistico. La superba grafica è accompagnata da un'ottima musica per ognuna delle fasi, il tutto è fornito di diversi effetti sonori.

Hostages

Parigi, domenica mattina, 3 febbraio 1988, 5.45 am.

Stava piovendo a dirotto e il capitano Cavendish era veramente imbestialito. Per l'ennesima volta alcuni terroristi, dei

veri figli di..., erano riusciti a guastare il suo weekend. Era assolutamente necessario salvare i cinque ostaggi tenuti prigionieri all'ambasciata. C'era solo una soluzione: inviare i suoi uomini del Gruppo d'Intervento della State Police Force all'assalto dell'ambasciata. Il gruppo era rapido, efficiente e ben preparato per sostenere con successo azioni di questa portata.

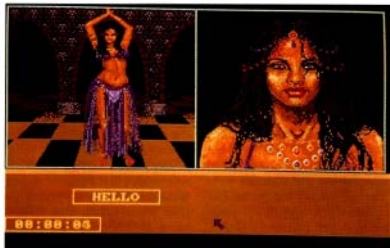
Le condizioni dettate dai terroristi erano assolutamente inaccettabili per il governo, perciò prese il megafono e iniziò a interloquire con i terroristi. Allo stesso tempo il suo cervello elaborava un piano d'azione strategico. Doveva posizionare tre uomini vicino all'ambasciata nei punti nevralgici e paracadutarne altri tre da un elicottero sopra l'edificio senza che i terroristi se ne accorgessero. I primi tre uomini avevano il compito di proteggere i secondi mentre si calavano con le corde del tetto per entrare dalle finestre dell'ambasciata.

Hostages è uno straordinario arcade game strategico. Lo scenario è originale e illustra le varie fasi strategiche e tattiche del gioco. Il giocatore comanda un gruppo d'intervento di sei uomini: suddiviso in due sottogruppi di egual numero, il primo sottogruppo è formato da eccellenti tiratori e il secondo da esperti scalatori. Si dovrà cercare di coordinare gli interventi in modo da riuscire a liberare gli ostaggi.

Questo è il primo gioco che vi permetterà di comandare simultaneamente sei personaggi sullo schermo. Non dovete assolutamente in nessun caso perdere questo software, anche se voi stessi siete appena stati presi come ostaggi!

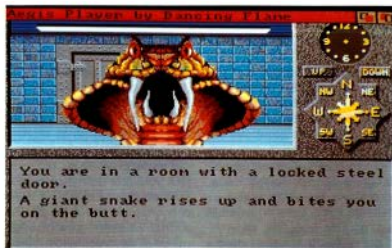
Return to Atlantis

Quella non era la guerra finale ma era altrettanto spiacevole. Durante la notte le bombe avevano raso al suolo gran par-

te della capitale e tra le macerie giaceva anche l'intera ala storica della biblioteca mondiale; dall'esplosione si sono potute recuperare delle statue acefale e pochissimi libri di storia il cui valore è ora inestimabile. È da sottolineare il rinvenimento, tra i vari reperti, di un manoscritto.

...durante questi secoli della storia umana gli oceani hanno rappresentato per l'uomo l'unico modo per sopravvivere, dal momento che aveva esaurito tutte le risorse dell'unico ambiente che era riuscito a dominare, la terra. Le colline inaridite, le foreste arse e i campi essicati erano i simboli di questa vittoria. Pure i deserti recavano le tracce dell'invasione industriale, il cancro del progresso alla fine dopo essersi esteso senza tregua era riuscito a cancellare ogni ogni vita sulla terra. Poi la terra fu spoglia, e l'umanità rivolse altrove il suo sguardo vorace: verso i mari.

Ora con la prospettiva di convivere in modo diverso con l'ambiente che lo circondava, l'uomo si adattò agli oceani. L'avidità dell'uomo fu sopraffatta dalla maestosità di quest'ambiente ed egli giurò solennemente che mai tali tesori sarebbero stati sfruttati in modo irrazionale e che le sue risorse dovevano essere preservate.

Poi il mare perse pian piano il suo alone di mistero, e vari furono gli uomini che furono disposti a non prestare più fede al giuramento fatto. Ben presto riuscirono a divulgare nel mondo intero le loro idee. Affermavano che la gestione dei massimi profitti era l'unico tipo di conduzione corretta, ma coloro che nel mare vedevano un'alternativa alla vita sulla terra non accettarono il ritorno a uno sfruttamento sfrenato delle risorse vitali.

La voglia di conservare questo nuovo modo di vivere per l'umanità portò alla riscoperta del mito di Atlantide, l'antico regno posto su un'isola dell'Atlantico, in cui uomini retti e generosi erano riusciti a vivere in pace. Si cercò così di ricostruire quella società anche se questo avrebbe portato a un confronto inevitabile tra le forze della terra e quelle del mare.

Ora la battaglia era iniziata e nello sforzo di mantenere l'opinione pubblica all'oscuro della gravità della situazione entrambi i contendenti avevano utilizzato la tattica del sabotaggio. Per evitare il dilagare del conflitto, la Fondazione sta unificando le forze della terra, reclutando gli uomini migliori usciti dall'Accademia.

Nel gioco voi siete un agente della Fondazione e dovrete cercare di mantere stabile l'equilibrio tra le due forze, proteggendo gli animali dell'oceano e combattendo tutti coloro che cercano di sfruttare le risorse marine in modo indiscriminato. E forse riuscirete a scoprire il segreto dell'isola perduta di Atlantide.

La base della Fondazione ha il suo quartire generale all'Isla Perdida, li otterete le istruzioni per le vostre missioni. Poi fate una puntata al Sea Thief Catè, è un locale turbolento, ma vi si possono reperire informazioni utili. Potrete rivolgere delle domande a vari informatori misteriosi, in seguito apparirà una mappa su cui è tracciata la rotta che dovrete seguire con la vostra nave. Arrivati a destinazione, la nave approderà e inizierete la vostra missione, aiutati dal computer di bordo (ART) il quale vi terrà costantemente informati sull'evolversi della situazione.

Appena caricato il programma vi verrà chiesto se avete già creato il disco di ID. Per fornire un'identità al vostro personaggio dovrete trovargli per prima cosa un nome che non superi i 17 caratteri, poi dovrete indicarne l'età e il sesso.

Ora potrete definire le caratteristiche di questo sommozzatore che sono nell'ordine: Psicologia, determina il comportamento dell'informatore nei vostri confronti, più siete bravi più l'informatore è disposto a collaborare con voi; Forma Fisica, determina la vostra abilità di nuotare e muoversi in generale sott'acqua; Telepatia, determina la distanza massima da cui riuscire a comunicare con il vostro robot; Autocontrollo, determina il consumo dell'ossigeno, più è elevato meno ossigeno viene consumato durante la missione subacquea; Programmazione, determina l'abilità con cui riuscite a controllare il vostro robot; Armi, determina l'abilità nell'uso delle armi; Salute, determina la condizione della vostra salute.

La salute influenza i vari attributi e se viene ridotta a zero dovrete cambiare identità e ricominciare nuovamente il gioco. Durante questa fase del gioco voi potrete incrementare di cinque punti un solo attributo a vostra scelta. Ora il computer vi chiederà di inserire il vostro dischetto ID nel drive interno (df0) per registrarvi il nuovo personaggio.

Dopo essere stati ragguagliati sugli scopi della missione dal vostro quartier generale per mezzo di una proiezione olografica, appariranno sullo schermo le sagome dei vari informatori. Se volete delle notizie sull'informatore con cui state trattando premete il tasto HELP, in questo modo apparirà un breve profilo dell'individuo. Per porre le domande potete operare in vari modi, ricordatevi che ognuno di loro è fornito di una personalità specifica. Dovrete perciò saper trattare, dal momento che tutti gli informatori dispongono di notizie importanti.

Innanzitutto salutateli (premere su HELLO) e quando avranno risposto al saluto scegliete uno dei modi per porre le domande. Potrete offrir loro delle bustarelle fino ad un massimo di cinque o supplicarli. Con alcuni potrete utilizzare anche le minacce se lo ritenete opportuno. Per rispondere alle domande premete su YES o su NO. Alla fine non vi resta che salutare premendo GOODBYE e ricordatevi che il tempo trascorso ad interrogare gli informatori vi verrà sottratto a quello a vostra disposizione per l'immersione.

Ora potrete recarvi sul luogo dove la vostra missione avrà inizio. Una volta approdati, comparirà sullo schermo la sezione della cabina della Viceroy e voi potrete scegliere se ottenere delle informazioni da ART che è il computer di bordo, visualizzare dei messaggi con MSG o por fine alla missione con BASE. Con il comando SCAN otterrete una prima scansione del fondo oceanico, e mediante le sue funzioni vi farete un'idea della zona che dovrete esplorare.

La vostra navicella è fornita di un Centro Medico mediante il quale potrete migliorare le vostre condizioni di salute. Nella stanza degli strumenti, posta sul ponte di comando, sono conservate le apparecchiature utilizzate per l'immersione. Nella vostra missione potrete portare con voi un massimo di cinque strumenti.

A questo punto siete pronti per immergervi e una volta immersi potrete notare sulla parte superiore sinistra dello schermo il tempo che vi rimane a disposizione e nella parte inferiore un menù per controllare i vari movimenti permessi sott'acqua. Potrete comunicare sia con RUF che con ART. Il menù vi permette di nuotare (swim); di effettuare delle scansioni (scan); di utilizzare la vostra arma (armed); di ritornare in superficie per rifornirvi di ossigeno (surface); di controllare il livello dell'aria (air) e la vostra salute (healt); di calcolare la vostra posizione (pos).

Durante la vostra immersione vi servirete delle molteplici funzioni di ART per portare a bordo gli oggetti rinvenuti, per fornirvi delle apparecchiature necessarie o per ottenere delle informazioni sull'ambiente. Per portare degli oggetti sull nave servitevi del comando Beam. Se ART dà una luce intermittente, vuole inviarvi un messaggio; per esaminarlo premete su Info. Per ottenere un profilo psicofisico del sommozzatore basterà premere Stats, e con Tools potrete servirvi degli strumenti a vostra disposizione.

RUF è il vostro aiuto a distanza sott'acqua, del quale vi servirete per allegerirvi almeno in parte del vostro lavoro. Anche RUF sarà fornito di una serie di comandi per utilizzarlo al meglio.

È possibile ripetere una missione già portata a termine, in questo modo potrete recuperare eventuali strumenti che vi servano per missioni differenti.

Al quartier generale della fondazione sono depositati i dossier; per facilitarvi il compito li abbiamo letti per voi.

Missione 1. The Golden Galleon:

Nel 1579 un uragano sorprese ed affondò il galeone spagnolo Tristeza, nel Mar dei Caraibi vicino ad Haiti. Assieme al galeone affondò uno scrigno pieno di monili d'oro sottratti agli Incas. La vostra missione consiste nel recuperare il tesoro, in modo che possa venir esposto nei musei del mondo invece di essere fuso e venduto al mercato nero.

Se fate una puntata al Caffè del Pirata potrete ottenere delle preziose informazioni. Ricordatevi che il proprietario, un bra-v'uomo, fu uno dei primi agenti della Fondazione.

Missione 2. The Weakest Link:

Nel Nord dell'Atlantico qualcuno ha reciso un cavo. Il cavo in questione sostiene la linea di collegamento tra le due super potenze. Entrambi i leader hanno interrotto le comunicazioni via satellite, dal momento che sospettano di essere intercettati. Ora senza collegamenti diretti i sospetti si ingigantiscono; dovete assolutamente riparare quel cavo prima che questo muro di diffidenza non porti a sacrificare delle nuove vite umane.

Missione 3. Of Pearls and Sunken Ships:

L'inquinamento del Mar del Giappone sta facendo morire tutte le perle e le scorte sparse nel mondo sono state distrutte dal diffondersi delle piattaforme petrolifere. Il prezzo delle perle sta salendo in tutti i mercati mondiali. Il vostro compito sarà di scoprire ed eliminare la sostanza inquinante.

Missione 4. Lost and Yet at Home:

Acquacity, la stazione sottomarina per le ricerche è stata colpita da un virus e la vita delle persone che vi lavorano è in pericolo. Il vostro compito sarà quello di recarsi nel Mar Mediterraneo e rintracciare il vaccino che si trova in qualche luogo nelle vicinanze della città ed infine portarlo agli abitanti di Acquacity.

Missione 5. And Only One May Live:

Da varie zone è giunta recentemente la notizia di avvistamenti di un celacantide che potrebbe trovarsi nella zona settentrionale del Canale del Mozambico. Stanno perciò raggiungendo il Magadascar frotte di pescatori di frodo con la speranza di catturarlo e venderlo per una cifra strabiliante a qualche collezionista. Il celacantide è stato alla fine catturato, narcotizzato e rinchiuso in una gabbia da uno di questi pescatori di frodo, ma durante uno scontro armato con altri pescatori la gabbia è caduta in mare. Dovrete perciò spostarvi fino a Grand Comoro, rintracciare il

celacantide narcotizzato e liberarlo.

Missione 6. Too Far Below:

Un piccolo sottomarino, impegnato in ricerche scientifiche è stato attaccato e danneggiato, gli uomini dell'equipaggio hanno trovato scampo nelle sfere di sopravvivenza che sono fornite di una riserva di ossigeno estremamente limitata. Il vostro compito è quello di rintracciarli e portarli in salvo.

Missione 7. Strange Allies:

Il CIEM, Consorzio Internazionale per lo Sfruttamento Minerario, ha ottenuto il permesso di effettuare degli scavi sul fondo dell'oceano per trovare e sfruttare eventuali depositi di manganese. Dovrete intervenire e questa volta avrete un alleato: la nave di Greenpeace che sta navigando in quella zona.

Missione 8. The Citadel Unsought:

In questa missione parteciperete attivamente allo studio scientifico concernente le rotte di migrazione delle balene. Dovrete recuperare dei dispositivi per la registrazione dei dati che alcune balene, oggetto di studio, si sono portate addosso per alcuni mesi.

Missione 9. The Hunter Trapped:

Un sottomarino americano, squarciato da un'esplosione, si trova sul fondo del Mar Baltico. Vi è stato richiesto di individuarne la posizione e riferirla alla nave di salvataggio Glomar Intimidator che si incaricherà del recupero dell'unità

Missione 10. The Arch and The Chain:

Nel Pacifico del Sud stanno scomparendo i krill e le creature che si nutrono di questi piccoli crostacei. La loro scomparsa porterebbe ineluttabilmente all'eliminazione di un anello fondamentale della catena alimentare marina. Dovrete individuare la causa ed eliminarla arrestando così questo letale processo.

Missione 11. Brothers of the Sea:

Il laboratorio di ricerca ha creato una versione migliorata del RUF, il RUF2. Ma durante il trasporto di questo congegno all'Isla Perdida, la nave che lo trasportava è scomparsa. Dovrete recarvi nell'Atlantico del Nord e cercare di ritrovare il RUF2.

Missione 12. Ghosts of Friends and Strangers:

In questa missione dovrete rintracciare un archeologo impazzito che cerca di dominare il mondo. Le indagini si svolgeranno nella zona delle isole Fiji.

Missione 13. When the Dead Shall Rise Again:

Tiwanaku, imperatore del Perù, vuole dominare il mondo e reintrodurre la cultura di Atlantide. Osservando la mappa troverete indicata la posizione di una città Inca inghiottita dalle acque davanti alle coste del Perù. Dovrete individuare questa città ed ottenere delle informazioni.

Missione 14. Return to Atlantis:

Anche in quest'ultima porzione di gioco la grafica tridimensionale accompagnata dagli splendidi effetti sonori vi offrirà un'esperienza subacquea di grande realismo.

Arazok's Tomb

L'archeologia è una professione affascinante. Un semplice colpo di piccone nella verde Scozia può riservarvi delle piacevoli sorprese ma può anche mutarsi in una terrificante avventura nel mondo dell'occulto. Nulla vi verrà in aiuto per risolvere il dilemma ed il fatto di essere uno dei migliori archeologi risulterà inutile. Ad ogni modo un altro membro della famiglia ci sarà indispensabile per attuare i nostri propositi. Dopo tutto, quando qualcuno dei parenti si trova nei guai chi credete che chiamino? Il cacciatore d'avventure! E ciò significa che chiamano proprio voi.

Con questo game la Aegis compie il primo passo per entrare nel mercato dei giochi, e il risultato è decisamente interessante. Se cerchiamo di etichettarlo, probabilmente lo potremmo chiamare 'avventura con grafica e testo'. Comunque il gioco non è del tutto perfetto anche se possiede varie configurazioni che possono sorprendere il giocatore. La grafica, composta da immagini di schermo statiche, è estremamente ben fatta. Lo stile grafico risente delle influenze stilistiche di Jim Sachs.

I giochi di questo tipo possono essere valutati per mezzo di due separate categorie che sono comunque collegate tra loro. I controlli meccanici del gioco e cioè come opera l'interfaccia utente, quant'è consistente il gioco, se opera in modo veloce e armonioso, se le figure sono più o meno utili, ecc. L'altra categoria valuta l'interesse della storia, se l'intreccio è ben costruito, se i disegni sono all'altezza, ecc. Osserviamo perciò il gioco sotto questi due aspetti.

La prima volta che caricate il gioco, la sequenza d'avvio vi chiederà la data e l'ora. Penserete che questo è un metodo inusuale per un gioco. Comunque, rispondete alla richiesta. Il programma utilizzerà i dati immessi per inizializzare un orologio presente sullo schermo. Quali siano i suoi scopi non è molto chiaro, dal momento che non l'ho utilizzato in seguito per il gioco.

Ma forse più avanti...

Accanto all'orologio, sono presenti sullo schermo vari gadget che faciliteranno l'esecuzione dell'adventure. Una specie di compasso vi permetterà di scegliere la direzione di marcia. Ci sono dei gadget separati per le direzioni di alto e basso. Una barra scroll vi permetterà di scorrere il testo all'indietro visualizzando in questo modo il testo già uscito dallo schermo. Questo vi sarà molto utile per rivedere cose dette precedentemente. La capacità di immagazzinamento non è infinita, perciò dovrete prendervi nota delle informazioni più importanti.

I menù standard di Amiga vi aiutano altrettanto bene, liberandovi dalle normali richieste del tipo: prendi, metti, guarda, leggi, ecc. Questa configurazione non è utilizzata al meglio, dal momento che non potete indicare un item sullo schermo e selezionarlo con una semplice pressione del pulsante. Non potete selezionare dal menù l'opzione GET e poi puntare sulla borsa piena d'oro e prelevarla per mezzo della pressione del pulsante; perciò non sarete mai completamente liberi dalla tastiera. Comunque anche se la flessibilità nei giochi è auspicabile non per questo il gioco perde il suo interesse. Un'altra opzione del menù è l'inventario, che è limitato a sette oggetti. Quest'opzione vi libera dal fatto di dover continuamente richiedere un'elenco degli oggetti in vostro possesso.

Per quanto riguarda l'analisi sull'altro versante, il gioco è un po' debole. Arazok's Tomb viene propagandato come un adventure per adulti e perciò ci si aspetta un prodotto di una certa qualità. Forse i film televisivi ci hanno abituato a ben altri livelli, ma delle espressioni del tipo, 'Il serpente ti strappa la gola' difficilmente sono etichettabili come espressioni da 'adulti' Durante le partite da me giocate (non sono mai arrivato al termine) non ho rilevato nella grafica alcuna scena violenta che potesse tenere lontani dal gioco i minori di 18 anni. Il game è perciò accessibile anche ai più giovani.

SMAU

Una rapida escursione nella maxi rassegna milanese dell'informatica

di Simone Concina

Dal 29 Settembre al 3 Ottobre si è svolta la 25ª edizione dello SMAU. Per l'occasione la struttura organizzatrice s'è davvero superata impostando una mostra indubbiamente enorme nelle sue dimensioni (30.000 mq. in più rispetto alla scorsa edizione), ma facilmente fruibile grazie alla distribuzione degli espositori per aree d'appartenenza.

Al momento, l'articolo viene scritto in data 1 ottobre, non siamo in grado di quantificare esattamente il numero dei visitatori, ma dalla concentrazione per mq., personalmente verificata durante la nostra visita, pensiamo che anche questo sia una cifra da record.

Come al solito, almeno negli ultimi anni, il caldo è insopportabile e tutti i visitatori, compresi i vostri umili cronisti, alla fine della giornata dimostravano chiari sintomi di afasia.

Se un suggerimento deve essere rivolto agli organizzatori, questo non può non essere quello di provvedere ad un sistema di condizionamento dell'aria all'interno dei padiglioni espositivi, così da evitar loro la necessità di premiare con medaglie alla memoria gli eroici visitatori, per non parlare degli espositori, che a migliaia ogni giorno percorrono i torridi ma popolatissimi sentieri della landa SMAU.

La Rocca di cristallo dell'Isola Misteriosa

Dopo aver visto un bel po' di cose più che interessanti, non seguendo il vecchio detto che consiglia di mangiare il boccone migliore alla fine del pranzo, ci siamo recati fiduciosi allo stand Commodore: forti anche delle voci che ce lo descrivevano dalle dimensioni accresciute rispetto all'edizione '77. In effetti il colpo d'occhio non era assolutamente male. Come un'isola bianca e azzurra con alla sommita' una Rocca in plexiglas, al centro di un atollo caratterizzato dalle stesse tonalità cromatiche, si ergeva la postazione della Com-

modore Italiana. Ai due lati, in luogo di nobili destrieri pronti alla tenzone, un'automobile Alpine da competizione e una moto da cross, credo, con sui fianchi i colori sociali della ditta d'oltre oceano a noi tanto cara.

L'immagine descritta è purtroppo molto più aderente alla realtà di quanto ci si possa immaginare, infatti ciò che appare all'occhio non è altro che una trasposizione metaforica di una realtà ben radicata: al centro una struttura fortificata che centellina i propri apporti e dintorno quella che è l'iniziativa spontanea di appassinati che finalizzano la propria passione per la macchina Amiga alle loro necessità commerciali. Proprio come dei vassalli all'ordine di un qualche Signore medioevale. Tale immagine, purtroppo, non ci restituisce solo la realtà italiana, ma pensiamo anche quella americana.

Poco ma vecchio

Le nostre più rosee speranze di trovare al padiglione Commodore alcune delle leccornie pregustate dai bollettini provenienti dal CEBIT e da altre mostre d'informatica sono subito state frustrate da un pannello murale, una bacheca in vetro, contenente le "novità" per Amiga commercializzate dalla Commodore Italiana. Tale scrigno, infatti, conteneva quanto segue (citiamo a memoria):

— Una scheda A 2301 o Genlock card da inserire direttamente in un Amiga 2000 Questa permette la miscelazione di immagini video provenienti da una fonte esterna purché il segnale sia di tipo composito.
— Scheda Janus II, sempre per il A 2000. Questa è l'evoluzione della Janus I e rende un Amiga compatibile AT MS-Dos. Infatti contiene un micoprocessore Intell 80286 a 8 MHz., permette il montaggio, come opzione, di un coprocessore matematico Intel 80287 e contiene un MByte di RAM. Non è consigliato richiederla, non è ancora disponibile.
— Scheda A 2620, con montato un processore Motorola MC68020 a 14.2 MHz. di Clock e un coprocessore matematico Motorola MC68881, oppure opzionalmente un MC68882. Inoltre, la scheda contiene 2 o 4 MByte di RAM e 64 KByte di ROM; è completamente autoconfigurante e particolarmente adatta a supportare il sistema operativo UNIX della AT&T: di cui peraltro non si è neppure vista l'ombra.
— Scheda A2090 e A2092. Hard Disk da 20 MByte e controller con interfaccia per dispositivi SCSI. Nella versione A2090 con A2094, l'Hard Disk ha una capacità di 40 MByte.
— Scheda A2058. Un'espansione di memoria per Amiga 2000 con 2 Mbyte di fast RAM incorporati ma con possibilità d'espansione fino a 4 o 8 MBytes.
— Scheda A 2060. Un modulatore video su scheda per A 2000 con uscite in composito, sia a colori che monocromatico.

Come si può vedere, forse qualcosa nell'elencazione l'abbiamo scordata, nulla di quanto è stato presentato può farci innalzare grida di giubilo. Infatti, delle nuove schede video, del sistema operativo UNIX, dei monitor ad alta risoluzione A2024 (1008 x 1024) senza interlace, dei transputer, dei nuovi custom, e di altro ancora non v'è ombra alcuna. Ad essere sinceri, e per non voler essere disfattisti oltre misura, v'è da dire che un responsabile della Commodore Italiana ci ha assicurato che alcuni prodotti facenti parte del secondo elenco, saranno distribuiti entro breve anche in Italia: prima della fine dell'anno. Non ci resta quindi che attendere e sperare in un fato più comprensivo per le nostre esigenze di poveri utenti Amiga; a quasi tre anni dall'uscita in Italia di tale splendida macchina. A tale proposito, sorvolo sulla profonda invidia provocata in noi da quanto esposto allo stand della Apple per il loro Macintosh II, mi pare, da poco più di un anno sulle scene. Ma come abbiamo detto, è meglio lasciar perdere.

Intorno alla Rocca con CTO

Per fortuna, il filo diretto che lega la Commodore Italiana alla C.T.O. di Bologna, non è servito a trasmettere a quest'ultima quell'apatia che sembra governare la prima. Infatti, i titoli visti allo stand della casa di distribuzione ufficiale Commodore, sono oramai moltissimi e, a quanto abbiamo saputo, gli stessi incontri propiziati dallo SMAU dovrebbero portare tra breve molti altri programmi al già ricco catalogo della CTO. Per quanto riguarda invece il software di sviluppo, vedi compilatori e tool di diverso tipo, abbiamo avuto assicurazione dal Dott. Madrigali che diversi contratti sono già andati a buon fine e che entro brevissimo tempo saranno disponibili i primi package.

Ciò che invece a nostro modesto parere non è stato ancora risollto, è l'oramai cronica assenza, dal mercato software per Amiga, di programmi professionali di buon livello. Tanto da far pensare che nè la casa madre statunitense, nè, di conseguenza, tantomeno le software-house di maggiore spicco hanno ancora deciso in quale fascia d'utenza collocare la pur versatile (forse un po' troppo?) Amiga. È infatti strano che oltre al Desk Top Video, dove effettivamente qualcosa di valido si stà proponendo, per esempio nel Desk Top Publiscing ci si ritrovi ancora al livello di un anno fa (di certo Page Setter non può essere una valida proposta a chi voglia usare Amiga per l'impaginazione elettronica professionale), mentre macchine come l'Atari, non certo superiori come prestazioni all'Amiga, propongono delle configurazioni davvero interessanti per il loro favorevolissimo rapporto tra qualità e prezzo. Pare infatti, quindi, che "... nessuno sappia cosa far fare da grande ad Amiga..." come, non senza una sfumatura malinconica nella voce, c'è stato riferito da un rappresentante della Commodore Italiana. Ciò che però noi vorremmo evitare di vedere, è che tale situazione porti gli utenti delusi ad un graduale ma irreversibile esodo verso altre macchine più supportate, rendendo così vano quel potenziale creativo che è un'indubbia prerogativa dell'Amiga. Ma questo è ancora un altro discorso che, come detto prima, non coinvolge certo solo la Commodore Italiana, anzi.

Una visita nell'arcipelago

Per contro, la situazione relativa gli importatori non ufficiali e ai produttori, sia di software che di hardware, di casa nostra, non è poi tanto male. Infatti, curiosando tra i vari stand delle ditte ospitate dalla Commodore Italiana, abbiamo potuto vedere dei prodotti che senz'altro possono far fronte alle esigenze di buona parte dell'utenza Amiga. Ciò, e secondo noi è ancora più importante, dimostra che un buon numero di operatori commerciali crede ancora nel fenomeno Amiga e si dà da fare affinché possa sopravvivere nonostante, appunto, il quasi disinteresse della Commodore stessa. D'altronde non ci dovremmo stupire poi tanto, visto che tale fenomeno è stato presente anche all'epoca del glorioso C 64.

Vediamo brevemente quali sono le proposte più interessanti.

La LOgiteK oltre a proporre i propri servizi, quali produzione di software e supporto ad attività di DTP e DTV, presentava una interessante linea di prodotti, sia hardware che software, tra cui i più imnteresanti sono:

— Gigatron. Espansione interna per A 500 e A 1000 da 1,8 MByte dotata di Clock interno ed autoconfigurante.
— Scanner manuale, che permette di rilevare immagini con 16 toni di grigi con

una risoluzione di 200 dpi, e di trasformarle in immagini IFF.
— La linea di Hard Disk Supra da 20 MByte a 250 MByte.
— Tavolette grafiche Kurta.
— Frame Buffer, un digitalizzatore a colori.
— Splitter, per trasformare un segnale composito in un segnale RGB.

La LOgitek srl ha sede in via Golgi 60, a Milano.

La Tecnodata, un'attivissima società con sede a Napoli in via C. Domenico Riccardi 347, progettisti tra l'altro del Videomaster 2995 venduto dalla Commodore — un sistema di elaborazione video semiprofessionale composto da genlock, digitalizzatore e con tre alloggiamenti per i drive A2010, con possibilità di gestione del segnale per mezzo di controlli del contrasto, colore, saturazione, dissolvenza —, si sono presentati con una linea di DTV dedicata sia ad usi professionali che amatoriali.
— Tec 500, genlock per A 500-1000-2000 con processore video banda passante da 5,5 MHz, controlli del segnale video in ingresso per colore/contrasto e luminosità, controllo della fase e controllo fade, mix level, croma key. Dotato inoltre di due ingressi e di un'uscita video composita.
— Amiga Splitter, un digitalizzatore con filtratura elettronica dei colori fondamentali e dotato di una commutazione computer-video che consente il monitoraggio continuo del processo di digitalizzazione.
— Tec 2500 Broadcasting, un genlock professionale per tutti i modelli di Amiga dotato di un frame buffer a colori da 1 a 50 immagini a colori programmabili e di 30 differenti tipi di effetti speciali disponibili.

La Bucolo Computer Graphics, si propone anche quest'anno come azienda impegnata nella produzione di prodotti eidomatici come video show e più in generale computer grafica. A tale proposito suscitano interesse alcuni titoli appartenenti ad una linea di software dedicata alle applicazioni televisive in diretta, denominata TeleAmigaLive.
— TotoLive, permette la gestione in diretta dei risultati del totocalcio, compresi gli aggiornamenti in schedina e automaticamente in classifica
— ElezLive, permette la gestione dei dati elettorali relativi ai singoli partiti, e ne effettua la comparazione con quelli di due precedenti consultazioni. I risultati vengono poi visualizzati in forma sia numerica che grafica per mezzo di due tipi divesi di istogrammi.
— WeatherLive, ottimo suporto alle previsioni del tempo in diretta grazie alla sua visualizzazione grafica di tutti i parametri connessi all'esposizione televisiva.
— Referlive, simile a ElezLive, supporta la gestione delle consultazioni referendarie, come i parametri voti, percentuali, sezioni e loro relativa espressione, in forma grafica con due istogrammi a forma di SI e NO in diretta.
— GameLive, consiste in un set di programmi dedicati a giochi di intrattenimento televisivo.

Oltre ai programmi descritti, la Bucolo propone dei Toolkit per arredare la casa e un package, denominato TeatrAmiga, atto all'automatizzazione delle funzioni di controllo tecnico necessarie in una rappresentazione teatrale, quali sincronismo audio e luci, quinte rotanti, scenari semoventi ed altro ancora.

La Atema di Firenze ripropone il suo oramai noto software dedicato alla creazione di tessuti. Il TEW, nella sua versione 2.02, ha subito ancora delle migliorie confermandosi, in tal modo, come un ottimo package professionale utilizzabile, non dimentichiamolo, su un sistema grafico che nonostante le sue capacità è disponibile a costi bassissimi.

Un altro prodotto molto interessante è stato visto nell'arcipelago, e anche se non espressamente progettato per Amiga, ha dimostrato subito una notevole adattabilità alle svariate esigenze di tale macchina: stiamo parlando dell'ultimo prodotto Polaroid, il Freeze Frame Video Image Recorder.

Il Freeze Frame è un hardware utilizzabile per trasportare tutto ciò che appare sul vostro monitor (hard copy), su una diapositiva a colori 24x36. E quando si dice tutto, si intende proprio tutto. Infatti, a differenza del Palette che poteva fermare solo le immagini prodotte dal computer (e spesso nemmeno quelle, vedi modo HAM), il Freeze Frame si collega direttamente al segnale del monitor, evitando quindi la mediazione del computer e di eventuali, e a quanto pare carenti, software di gestione. In questo modo, se per mezzo di un genlock si immettessero delle immagini da un video registratore e vi si sovrapponessero delle altre immagini prodotte per mezzo di un qualsiasi programma grafico, l'insieme completo delle immagini da noi prodotte verrebbero fermate su una splendida diapositiva a colori. È importante specificare che le immagini non devono essere necessarisamente statiche. Infatti, il Freez Frame quando viene attivato, permette di fermare qualsiasi immagine, anche in movimento quindi, in una memoria di quadro e di poterne disporre poi a piacimento.

Non è certo questo il momento di dare una descrizione approfondita di questa apparecchiatura, ci riserviamo di farlo al più presto, ne daremo quindi solo una breve descrizione.

L'apparecchio è composto da due parti, una console di controllo e la camera oscura. La prima ospita quelli che sono i comandi di gestione : regolazione luminosità della fotografia, regolazione contrasto della fotografia, pulsante di conferma esposizione, aumento della risoluzione, selezione pellicola fotografica. Il parallelepipedo nero che abbiamo chiamato camera oscura, oltre a contenere il monitor, la circuiteria e la macchina fotografica, sul suo pannello posteriore mette in mostra una vera e propria collezione di connettori atti a rendere questa aparecchiatura veramente versatile. I connettori sono: controllo automatico dorsi, collegamento console, uscita analogica RGB, ingresso analogico RGB, ingresso sincronismo esterno, uscita sincronismo esterno, uscita monitor supplementare, ingresso RGB in TTL e uscita RGB in TTL. Inoltre, quattro switch controllano: selezione NTSC/RGB, selezione RGB TTL/Analogico, selezione impedenza terminale e selezione di sincronismo esterno o interno.

Per riassumere un buon prodotto che di certo non mancherà d'incuriosire coloro i quali si interessano di produzione grafica con Amiga. Dimenticavamo il prezzo: veramente alla portata di molti; telefonare alla Polaroid per credere.

Conclusioni

Come abbiamo potuto vedere, nell'arcipelago bianco azzurro le cose allettanti non mancano di certo, ma ciò che ci è stato confermato è preoccupante. Infatti, anche in queste splendide acque serpeggia il malcontento nei riguardi di chi troppo spesso si chiude nel proprio castello senza dare ascolto a chi per passione, e anche perché no per interesse, vorrebbe una collaborazione più stretta e più dinamica tra le varie parti. Parti, non dimentichiamocelo, parimenti interessate allo sviluppo di una macchina e quindi di un mercato che diversamente rischia la morte per immobilismo. Se poi i diretti interessati non ci avessero pensato, noi siamo convinti che quanto detto porterebbe a delle ripercussioni finanziarie non certo di poco conto: non crediamo infatti che l'utenza Amiga, sia pure composta da veri appassionati, sarebbe disposta a soprassedere su un ennesimo fallimento Commodore, quindi...

COME INTERPRETARE E TRADURRE LE STRUTTURE DEL SISTEMA OPERATIVO

di Mr. Lambda

Dovrebbe essere risultato chiaro fin dal primo nostro incontro che per manipolare proficuamente il sistema Amiga è necessaria un'adeguata conoscenza del suo sistema operativo e delle sue strutture in particolare. In questo articolo vi presenteremo come tradurre in Forth le strutture di sistema e come interpretare e utilizzare in Forth i file include del C e dell'Assembly.

Innanzitutto il dilemma: Forth versus C ?!

Allo stato delle cose è indispensabile per il serio programmatore in Forth una conoscenza perlomeno sufficiente del linguaggio C. Uno dei più importanti adattamenti, infatti, che i programmatori Forth devono fare quando programmano suii'Amiga è l'interfaccia Forth/C.

Poiché il sistema operativo dell'Amiga è stato scritto in C e sarebbe quantomeno folle tentare di sviluppare applicazioni per la macchina senza utilizzare le immense risorse messe a disposizione dal kernel ROM, sono necessari e inevitabili alcuni compromessi, che peraltro, senza tradire lo spirito della programmazione Forth, dimostrano la estrema versatilità del linguaggio e il penetrante realismo del programmatore formato a simile scuola.

Uno dei meccanismi più importanti che deve essere implementato è quello che permette di effettuare le chiamate al kernel, e di esso abbiamo presentato adeguati esempi negli incontri precedenti. Però si sarà ormai capito che per effettuare tali chiamate ci sono anche un grande numero di strutture dati da definire e inizializzare. Noi pensiamo sia giunto il momento di dedicarci approfoditamente a questo argomento considerando attentamente le traduzioni in Forth delle strutture e delle costanti utilizzate in C e in Assembler. Naturalmente continueremo a servirci del Multi-Forth della Creative Solutions per i nostri esempi, ma la situazione con qualsiasi altro Forth presenta approssimativamente le stesse caratteristiche.

I tipi

Probabilmente la più grande differenza tra le strutture implementate in C e le loro definizioni equivalenti in Multi-Forth è la mancanza in Multi-Forth della varietà di tipi presente invece in C. Consideriamo, per esempio, la struttura utilizzata per controllare lo sprite hardware in C:

```
struct SimpleSprite
{
 UWORD *posctldata;
 UWORD height;
 UWORD x;
 UWORD y;
 UWORD num;
};
```

Il primo membro della struttura, posctldata, è definito come un puntatore di tipo UWORD. Esso indirizzerà il sistema ad una struttura SpriteImage che contiene due word senza segno di sedici bit di dati per il controllo della posizione e l'immagine

in bit dello sprite. Il membro successivo, height, è un valore di sedici bit senza segno che specifica la dimensione verticale di uno sprite in linee di visualizzazione non interlacciata. Gli altri membri sono word senza segno che contengono le coordinate dello sprite e il suo numero.

L'asterisco che precede il primo membro e che denota un puntatore in C viene utilizzato dal compilatore, insieme al tipo dichiarato di dato, per impedire al programmatore di puntare a qualcosa dalle dimensioni errate. Dal momento che posctldata è qualificato da UWORD questo campo può venire utilizzato solamente per puntare a dati di quella dimensione, cioè a word di sedici bit senza segno.

Dal momento che coloro che programmano in Forth avranno spesso a che fare con l'Assembler, consideriano un attimo le soluzioni offerte dal file include relativo in Assembly:

```
STRUCTURE  SimpleSprite,0
APTR       ss_posctldata
WORD       ss_height
WORD       ss_x
WORD       ss_y
WORD       ss_num
LABEL      ss_SIZEOF
```

E ora confrontiamo la nostra struttura in C e in Assembly con la definizione equivalente in Multi-Forth:

```
structure SimpleSprite
      ptr: + ssposctldata
   short: + ssheight
      short: + ssx
      short: + ssy
      short: + ssnum
structure.end
```

Non c'è alcun controllo sul tipo di dati (type checking) in Multi-Forth (come pure in Assembly), e neppure nel linguaggio Forth in generale è previsto un simile controllo, la responsabilità del controllo del tipo di dati è sempre demandata al programmatore. In generale, qualsiasi entrata in una struttura C che contiene un asterisco può venir definita 'ptr:' in Multi-Forth. La parola (word) 'ptr:' specifica che siamo in presenza di un membro di 32 bit e crea il nome del membro. Ma c'è di più. I nomi dei membri in Multi-Forth non sono dipendenti dal contesto come i membri di strutture o unioni in C. La conseguenza di ciò è che in Multi-Forth ciascun membro di una struttura deve possedere un nome unico; non può condividere un nome con un membro di un'altra struttura, oppure con qualsiasi altra parola a meno che queste non risiedano in un vocabolario separato. Questa è una delle ragioni per aggiungervi un prefisso, un postfisso o un altro segno di riconoscimento. Creative Solutions suggerisce l'utilizzo di un segno di somma '+' per i nomi dei membri dal momento che il segno di somma denota il loro comportamento in run time, cioè in tempo esecuzione, e li differenzia dalle altre parole Forth. In questo caso 'ss' si riferisce al nome della struttura SimpleSprite. Creative Solutions ha anche scelto di utilizzare 'short:' per indicare un valore a sedici bit perché 'word' significa (e si referenzia a) qualcosa d'altro in Forth.

I membri delle strutture, dunque, possono solamente assumere tre tipi in Multi-Forth. E questi possono essere valori a 8 bit, a 16 bit oppure a 32 bit. Non c'è neppure alcun controllo per valori con segno e senza segno, oppure se un campo contiene un indirizzo oppure un valore. Da un punto di vista funzionale, un membro 'long:' e un membro 'ptr:' sono identici. Ciascuno di essi aggiunge quattro byte alla dimesione della struttura, e nomi differnti vengono utilizzati solamente a scopo di chiarezza. Nello stesso modo, WORD, UWORD, SHORT, USHORT e qualsiasi altro valori di quel tipo in C denotano un valore a 16 bit che può venire sostituito con 'short:' in Multi-Forth. E ancora, in Multi-Forth qualsiasi campo a 8 bit può essere indicato con 'byte:'. Per tutto questo si può dire che per quanto riguarda i tipi il Forth si avvicina molto al trattamento dei tipi come viene sviluppato in Assembly. Per un confronto si consideri per esempio il file include exec/types.i.

Array

Gli array all'interno delle strutture vengono gestiti in Multi-Forth con forme al plurale delle parole già menzionate. Ed eccovi un esempio che può rendervi chiare le cose appena dette:

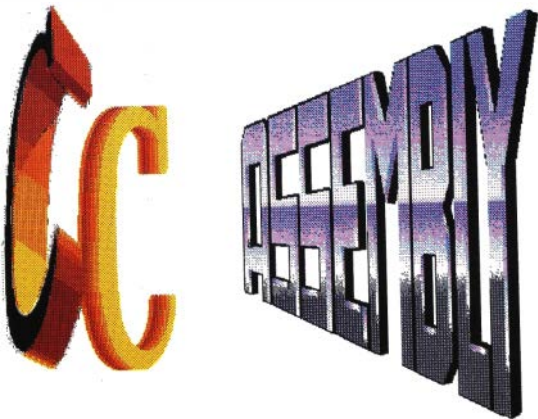
```
struct collTable
{
 int (*collPtrs[16]) ();
};
```

Questa struttura è presente nel file include in C graphics/gels.h. Come avrete capito contiene un array di 16 puntatori a 16 indirizzi di procedure per la gestione della collisione. Vediamo la versione corrispondente in Assembler:

```
STRUCTURE collTable,0
LONG cp_collPtrs,16
LABEL cp_SIZEOF
```

Ed ecco l'equivalente in Multi-Forth:

```
structure collTable
  16 ptrs: + ctcollPtrs
structure.end
```

Ricordate, come abbiamo già detto precedentemente, che ogni entrata in un campo di una struttura in C che contiene un asterisco può essere definita con 'ptr:'; e ammettiamo pure che ciò ne semplifica un po' la sintassi. Naturalmente abbiamo anche perduto qualchecosa: infatti, dovremo calcolare l'offset corretto all'entrata del

campo appropriato quando utilizzeremo successivamente la struttura, invece di specificarne semplicemente l'indice. Sotto questo punto di vista, questo è un cattivo esempio. In un applicazione reale, noi non utilizzeremmo proprio questa struttura, sostituendola con un metodo semplice di indicizzazione in un array, con una sintassi simile a

```
8 collTable @
```

per prelevare l'indirizzo appropriato. Seguire pedissequamente il C può portare, a volte, a soluzioni non ottimali.

Ma procediamo. E vediamo un altro esempio che utilizza una struttura per un file information block. In C:

```
struct FileInfoBlock
{
  LONG    fib_DiskKey;
  LONG    fib_DirEntryType;
  char    fib_FileName[108];
  LONG    fib_Protection;
  LONG    fib_EntryType;
  LONG    fib_Size;
  LONG    fib_NumBiocks;
  struct  DateStamp fib_Date;
  char    fib_Comment[116];
};
```

E come sempre cosideriamo la traduzione in Assembler:

```
STRUCTURE FileInfoBlock,0
   LONG      fib_DiskKey
   LONG      fib_DirEntryType
   STRUCT    fib_FileName,108
   LONG      fib_Protection
   LONG      fib_EntryType
   LONG      fib_Size
   LONG      fib_NumBiocks
   STRUCT    fib_DateStamp,ds_SIZEOF
   STRUCT    fib_Comment,116
   LABEL     fib_SIZEOF
```

E il suo equivalente in Forth:

```
structure FileInfoBlock
              long: + fibDiskKey
              long: + fibDirEntryType
          108 bytes: + fibFileName
              long: + fibProtection
              long: + fibEntryType
              long: +fibSize
              long: + fibNumBlocks
 DateStamp struct: + fibDate 116
             bytes: + fibComment
structure.end
```

In questo esempio noi abbiamo introdotto la parola 'struct:' ed essa merita senz'altro qualche spiegazione.

Struct e union

In tempo esecuzione (run-time), il nome di una definizione di struttura (structure template) ritornerà la dimensione della struttura stessa sullo stack. Tuttavia, l'azione in tempo compilazione (compile-time) dell'espressione:

```
DateStamp struct: + fibDate
```

—è semplicemente di creare la parola +fibDate e aggiungere la dimensione della struttura DateStamp alla struttura FileInfoBlock.

Il Multi-Forth fornisce anche un sinonimo per la parola 'bytes:' che in questo caso può essere preferibile. Potete quindi sostituire 'string:' a 'bytes:' senza che si produca alcuna differenza di azione.

La sintassi del Multi-Forth incomincia a distinguersi nettamente quando si accinge a tradurre le definizioni di unioni, che la Creative Solutions preferisce chiamare 'multiforme'. Noi utilizzeremo come esempio la struttura MemEntry, utilizzata per allocare memoria e presente nel file include exec/memory.h. Essa dimostra bene l'inserzione di un unione all'interno di un'altra struttura dati.

```
struct MemEntry
{
union {
    ULONG   meu_Reqs;
    APTR    meu_Addr;
    }
    me_Un;
    ULONG   me_Length;
};

#define me_un       me_Un
#define me_Reqs     me_Un.meu_Reqs
#define me_Addr     me_Un.meu_Addr
```

Ed ecco la versione corrispondente in Assembly, ma semplificata:

```
STRUCTURE ME,0
    LABEL     ME_REQS
    APTR      ME_ADDR
    ULONG     ME_LENGTH
    LABEL     ME_SIZE
```

La struttura potrebbe essere definita in Multi-Forth eliminando, come abbiamo appena visto nella versione in Assembler, la necessità del riferimento ai sinonimi presenti in C, in questo modo:

```
structure MemEntry
   one.of    long: + meReqs
    or.of      prt: + meAddr ;go.on
             long: + meLength
structure.end
```

Questo esempio vi fornisce un buon esempio dell'utilizzo delle multiforme in Multi-Forth. E ora passiamo alla discussione della traduzione degli statement #define del C in Multi-Forth.

#define e altro

Il C utilizza di solito i #define per la definizione di costanti simboliche o manifeste. Uno statement come

```
#define CMD_INVALID 0
```

può essere tranquillamente tradotto come

```
0 constant CMD_INVALID
```

Nei file include in Assembly in questo caso si trova utilizzata la macro DEVCMD, in altri casi la direttiva EQU.

Una costante stringa come

```
#define EXECNAME "exec.library"
```

che in Assembly viene trattata come una macro:

```
EXECNAME macro
      dc.b 'exec.library',0
      ds.w 0
      endm
```

può essere definita in Multi-Forth come:

```
0" exec.library" constant EXECNAME
```

Ma non tutti casi sono così semplici. La chiave per la comprensione di questa parola chiave è ricordare che #define è una direttiva del preprocessore. Essa sostituisce istanze di una stringa con un'altra prima che il codice sorgente sia passato al compilatore, e qualsiasi stringa è valida, non solamente valori numerici. In alcuni casi potrebbe essere appropriato tradurre gli statement #define in definizioni che iniziano con i due punti (colon definition).

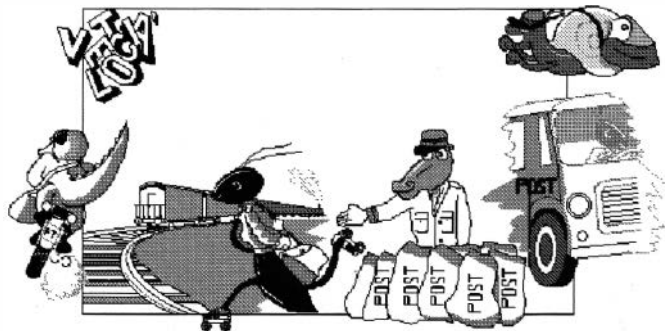
Capire per inventare

In generale non è troppo difficile tradurre i file include del C e dell'Assembler in Forth. E potrebbe anche sorprendervi quanto sia facile, qualche volta, tradurre del codice ad alto livello dal C al Multi-Forth. Nei numeri precedenti ve ne abbiamo dato una dimostrazione con le funzioni CreatePort e DeletePort che corrispondono ai listati presenti nel ROM KERNEL EXEC MANUAL, queste versioni potrebbero però essere migliorate suddividendole in diverse parole per renderle più comprensibili e più pratiche da utilizzarsi. La traduzione letterale del C, perciò, non è sempre la miglior cosa da farsi per realizzare le medesime cose in Multi-Forth. Dopo aver capito il problema ed essersi familiarizzati con gli strumenti che il sistema di Amiga offre, conviene seguire una strada propria; e non è questo poi lo stile Forth?! Molte cose comunque si imparano dall'osmosi tra i due linguaggi.

CORRISPONDENZA

La rubrica ''Corrispondenza è stata voluta al fine di definire uno spazio, interno alla rivista, dedicato al confronto e al contributo di idee, provenienti della galassia dell'utenza Amiga

Per avervi accesso, inviate le vostre missive a:

Spett. redazione
''Amiga Magazine'' rubrica ''Corrispondenza''
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 MILANO

Un eccezionale strumento per la manipolazione delle immagini

Molti disegni o digitalizzazioni creati con i vari programmi ad essi dedicati, spesso non sfruttano pienamente le possibilità grafiche che l'Amiga è in grado di fornire. Partendo da questo presupposto è nata l'idea di trovare un mezzo in grado di gestire, in modo professionale, le immagini Amiga. Naturalmente chi poteva essere la persona più adatta a programmare qualcosa del genere, se non quel geniaccio di Justin V. Mc Cormick il quale, a sua volta, ha venduto il lavoro alla Progressive Pheripherials & Software, Inc. di Denver. Purtroppo questa meraviglia rischiava di rimanere sconosciuta a noi italiani, infatti, spesso, programmi di notevole interesse sono destinati a circolare solo in modo underground e, ciò che è peggio, senza alcuna manualistica. Per nostra fortuna invece la C.T.O. ha pensato di acquistare PIXmate per il mercato nazionale e di tradurre il manuale allegato.

Và subito detto che PIXmate non è un programma ideato per la creazione di immagini ma "solo" per la gestione di esse. Non fatevi ingannare da quel -solo- scritto tra virgolette, poiché in realtà le possibilità offerte da questo tool sono infinite. Forse a qualcuno di voi sarà capitato di scattare delle fotografie con un po' troppa noncuranza, e di verificare al momento dello sviluppo, che queste ne risultassero sovraesposte o sottoesposte; come fare per recuperarle? Solo le magie di un abile addetto ai lavori possono, in questi casi, fare qualcosa, e non sempre. Con PIXmate invece potete gestire delle immagini digitalizzate in modo estremamente facile, agendo sulla luminosità e l'intensità dei colori in modo da ottimizzare il risultato finale. Infatti il programma comprende il più importante elemento per la valorizzazione delle immagini che consiste nella capacità di distinguere nell'intera schermata ciascun pixel e riconoscerne i diversi valori. Questa particolare e dettagliata lettura permette di selezionare minuziose parti di un immagine anche molto complessa.

Forse la gestione del palette è ciò che esalta in modo più appariscente le qualità di questo programma, la possibilità di agire su ciascun colore alterandone la luminosità e l'intensità permette un'infinita gamma di variazioni sulla stessa figura. Come maggior parte dei programmi anche questo usa i comodissimi menu a discesa e le pratiche abbrevazioni da tastiera. Una parola di riguardo merita il manuale allegato al programma, esso vi guida passo dopo passo negli aspetti più reconditi di questo complesso, ma estremamente comprensibile, universo grafico. Una esplorazione a freddo delle varie opzioni elencate nei menu non vi servirà a molto, mentre invece è necessario armarsi di un po' di pazienza e seguire le indicazioni del manuale, ma sopratutto è fondamentale sperimentare i vari esempi in esso indicati. Se seguirete questo metodo siamo sicuri che resterete letteralmente strabiliati dai risultati, come del resto è successo a noi.

Entriamo nei dettagli

Per prima cosa il manuale indica i modi di caricamento di PIXmate e come effettuare delle copie del dischetto per l'uso quotidiano. Purtroppo qui incontriamo una procedura alquanto fastidiosa, l'unica dell'intero programma. Per motivi di Copyright il dischetto originale contiene una pro-

di Alessandro Prandi

tezione che ne permette la duplicazione solo a determinate condizioni. La procedura da seguire è alquanto inconsueta, vediamola un attimo:

Per prima cosa bisogna trasportare i file "PIXmate" e "PIXmate.DATI" in un disco Workbench. A questo punto sarebbe ovvio pensare che tutto sia finito, invece no, ora viene il bello. Ottenuta la copia di PIXmate, la dovete inserire in DF0:, quindi inserire l'originale in DF1:, digitare Pixmate:Pixmate da CLI e, a caricamento ultimato, estrarre la copia originale da DF1:.

Ne risulta in pratica che ad ogni uso di PIXmate vi sarà indispensabile avere a portata di mano il dischetto originale, oltre alla copia. Intuibili i risvolti negativi.

Passiamo oltre a questo neo e vediamo invece le caratteristiche propriamente grafiche del programma. I menu disponibili sono cinque e tutti molto ricchi. Nel menu Progetto troviamo oltre alle solite istruzioni di Carica, Salva ed Esci la voce Comprimi che consente di attivare o disattivare la compressione dei file IFF. Altre interessanti opzioni di questo menu sono Leggi e Scrivi, questi due comandi permettono invece di caricare o salvare formati diversi dall'IFF. Dopo il menu Progetto troviamo quello di Modifica o Edit. In questo menu sono contenute le varie opzioni di editing dello schermo: possibilità di passare da uno schermo all'altro, PIXmate ne gestisce ben due: trasferimento dell'immagine dal primo schermo al secondo, selezione di aree specifiche dell'immagine e loro eventuali spostamenti. Alla fine troviamo l'opzione Cattura DPaint, il programma cercherà eventuali immagini DPaint in memoria e le trasferirà in PIXmate per eventuali modifiche.

Naturalmente il menu più complesso è quello dei Colori, proviamo ad analizzarlo. Alla prima riga c'è l'opzione Ciclo che effettua la rotazione dei colori di un'immagine. Quindi troviamo il Palette che ci consente una varietà di funzioni di modifica e correzione dei colori, da qui si parte per i primi ritocchi ad un'immagine. La finestra per la regolazione dei colori, o BIAS, agisce sul contrasto, la saturazione e l'intensità dei colori, offrendo inoltre, il livello delle gradazioni di rosso, verde e blu. Sicuramente l'opzione maggiormente utilizzata di questo menu è quella di Compatta. Con questo comando si riordina la ColorMap per intensità, si eliminano tutti i colori non adoperati nell'immagine ed inoltre si assicura che ogni colore possa essere usato una sola volta. I tre compiti appena descritti risultano di grande aiuto durante la fase di perfezionamento della figura. Questo processo elimina inoltre tutti i colori

superflui dall'immagine e vi permette di ridurre il numero di bitplanes.

Un altro trucco che potete usare è quello della riduzione dei colori, in questo modo si può diminuire il numero di colori di un'immagine. Questo lo potete usare per molteplici scopi inclusa la riduzione dei bitplanes.

Una delle peculiari caratteristiche del programma PIXmate consiste sicuramente nell'estrema semplicità con cui si possono convertire immagini da un formato all'altro. Convertire una figura dal formato HAM (Hold and Modify) nel formato a 32 colori o viceversa risulta straordinariamente semplice e sorprendentemente efficace. I risultati si possono definire, senza timori di smentite, ottimi. Un'altra conversione possibile è quella da HAM a EHB (Extra-HalfBright); il modo EHB sopporta 64 colori sullo schermo. Va ricordato che i primi modelli Amiga 1000 non gestiscono il modo EHB.

Il menu Effetti contiene invece le più intricate e allo stesso tempo potenti applicazioni di PIXmate. Il primo termine indica il formato dello schermo e consente la scelta dei bitplane da visualizzare. Da questa finestra inoltre potete selezionare il formato della bitmap (da 320 × 200 a 736 × 480) e il tipo di schermata, lo-res, hi-res, interlace, 16, 32 o 4096 colori.

Voce ancor più importante è quella di Elabora; le sue applicazioni si dividono in tre categorie: pixel, matrice e di tipo logico. PIXmate può ricavare dei dati per queste operazioni sia dall'immagine corrente che da quella depositata nell'altro schermo o da una combinazione delle due.

L'applicazione pixel scruta l'intera immagine e cambia solo i pixel che ricadono nei parametri da voi preindicati. Le operazioni matrice invece cambiano tutti i pattern di pixel come da voi selezionato. Le operazioni logiche invece agiscono su tutta l'immagine, ed hanno una funzione di filtro.

Vorremmo dilungarci e specificare ulteriormente le varie interconessioni possibili tra tutti questi comandi ma purtroppo ci servirebbero ancora molte pagine e rischieremmo di confondervi le idee. Invece, senza alcun dubbio, possiamo affermare che questo package è il primo prodotto veramente completo che racchiude dei potenti e veramente preziosi tool in un unico programma.

Se siete interessati alla computer grafica su Amiga e non avete ancora comperato questo programma, vuol dire che non siete al passo con i tempi, se invece già lo possedete, purtroppo significa che per l'ennesima volta siamo arrivati noi troppo tardi.

CORSO COMPLETO IN AUTOISTRUZIONE ALL'USO DEL PERSONAL COMPUTER

PC MASTER

Questo corso in autoistruzione, fornisce al lettore tutte le conoscenze necessarie per permettere di utilizzare efficacemente dBase III e dBase III plus, i più noti e diffusi data Base. Il corso è strutturato in due momenti integrati tra loro: testo e software interattivo.

Il testo, con metodologia semplice e graduale, guida il lettore ad una completa comprensione e padronanza dei concetti fondamentali, permettendo l'apprendimento anche a coloro che non hanno ancora acquisito una preparazione specifica sull'argomento.

Il software, simulando le caratteristiche e le situazioni operative del dBase III e dBase III plus, permette di esercitarsi immediatamente sugli argomenti trattati, fornendo in tal modo quella interazione pratica indispensabile all'apprendimento.

Al fine di rendere la trattazione più esaustiva possibile è prevista una SEZIONE ARGOMENTI, in cui vengono trattati temi teorici relativi alla gestione delle BASI DI DATI, non strettamente legati all'uso del dBase III, ma lo cui conoscenza è utile per una comprensione dei criteri più generali che stanno alla base della organizzazione degli archivi.

DISPONIBILE NELLE VERSIONI 5¼" e 3½"

Con il 1° fascicolo, troverete i dimostrativi dei programmi, che permettono di analizzare in dettaglio tutti i comandi e le possibilità di questi pacchetti.

PIU' DEMO dB III plus

ALL'INTERNO DEL CLI: PARTE TERZA

Continua il viaggio nell'AmigaDOS

di Alessandro Prandi *Parte terza*

Nelle puntate precedenti abbiamo sbirciato all'interno del CLI, ora invece prenderemo in esame alcune utility AmigaDOS per la manipolazione dei file.

Sicuramente nelle vostre esplorazioni del disco Workbench sarete incappati nella directory C, ricolma di comandi fino all'inverosimile. A noi, per il momento, interessano in particolare due di questi comandi: SEARCH e SORT. Queste sono due utility dedicate all'esaminazione dei file. Search cerca le stringhe testo mentre SORT mette in ordine alfabetico un listato.

Ci sono due modi per cercare una stringa testo tra un gruppo di file; o fissare attentamente lo schermo mentre il contenuto dei file vi scorre davanti. E, forse, non ne vale la pena, oppure lasciare questo compito ad Amiga indicandogli appunto il comando SEARCH. La ricerca è uno di quei tediosi lavori in cui I computer, al contrario di noi, eccelle. Gli scrittori vogliono localizzare citazioni o frasi troppo usate. I programmatori possono voler trovare una particolare subroutine dimenticata nelle risme di codice sorgente.

Per iniziare una ricerca di una stringa dovete scrivere

?	trova ciascun carattere;
#<p>	trova zero o più ripetizioni di <p>;
<p1>!<p2>	trova o <p1> oppure <p2>
()	delimita un pattern

Per esempio:

Anno?	trova Anno1, Anno2 eccetera
Anno#?	trova Anno, Anno1, Anno2, Anno12 eccetera
Anno(1!2)	trova Anno1 e Anno2

Dopo il pathname mettete la stringa da cercare. Se questa è composta da due o più parole, dovete racchiuderla tra virgolette. Ricordate inoltre che SEARCH non riconosce il maiuscolo/minuscolo, pertanto "MAGAZINE" e "magazine" verranno trattati allo stesso modo.

Se aggiungete la parola ALL provocherete una scansione completa dei file in tutte le sottodirectory.

Man mano che Search procede nel suo compito esso visualizza ciascun nome di file. Durante l'arrivo dei dati esso stampa il numero di linea e la riga di testo contenente la stringa. Come per tutti i comandi CLI, potete reindirizzare l'output in modo da salvarne quello risultante dalla ricerca, in un file da usare successivamente per diverse applicazioni. Per esempio, per immagazzinare il risultato di una ricerca in un file chiamato RicercaStringa, dovete digitare SEARCH>RicercaStringa eccetera.

Per interrompere il comando SEARCH premete CTRL/C. Con CTRL/D invece si arresta la ricerca nel file corrente per riprenderla al file successivo. A proposito, SEARCH manipola i file testo e non, pertanto potrete anche effettuare la scansione dei programmi.

Quando avete trovato ciò che cercavate, potreste desiderare di classificarlo con ordine. L'AmigaDOS è dotato di un semplice selezionatore chiamato SORT. Con SORT le righe di un file verranno posizionate in ordine ASCII ascendente: prima la punteggiatura, poi i numeri e infine le lettere. Anche questo comando ignora il MaiuScOlO/minUsCoLo. Scrivete:

SORT <fileentrata> <fileuscita> (COLSTART <n>)

Questo comando riordina i file prescelto (fileentrata) e produce un nuovo file (fileuscita). Se volete indirizzare l'output sullo schermo invece che in un file, basterà specificare "*" per <fileuscita>.

Il SORT inizia la comparazione dalla prima colonna o dalla colonna <n> se avete specificato la COLSTART (colonna dalla quale si vuole iniziare il riordino). Se possedete una lista del tipo:

Roberto	68445
Andrea	76866
3450 TV	xxxxx
Sara	66654

Dando il SORT ad un file come questo otterete una lista in ordine alfanumerico:

3450 TV	xxxxx
Andrea	76866
Roberto	68445
Sara	66654

Se il SORT è fatto con il COLSTART 22, il file di output risulterà riordinato con il codice ZIP:

Sara	66654
Roberto	68445
Andrea	76866
3450 TV	xxxxx

Sort legge l'intero file nella memoria e lavora da lì. Se non c'è abbastanza spazio nella RAM il sistema andra in tilt. Assicuratevi che le dimensioni dello stack siano sufficienti per contenere il file d'entrata. Per verificare le dimensioni del file scrivete:

LIST <file>

e quindi dimensionerete lo stack con:

STACK <dimensione>

Un file di qualche migliaio di righe impiega diversi minuti per essere riordinato. Sort sembra adoperare una decente tecnica di classificazione, poiché il tempo adoperato dall'operazione è proporzionale a quanto sta n per registrare n, dove n è il numero delle righe. Infatti n che registra n è il limite teorico della velocità di riordinamento.

SEARCH e SORT vi sono dati assieme all'Amiga, ma oltre a queste due utility di indubbio valore ce ne sono delle altre di pari, se non superiore, livello.

Dopo aver letto questo articolo passate alla sezione della rivista Disco Magazine e troverete alcune righe dedicate a un programma di inestimabile valore (il file eseguibile si trova nel dischetto), il nome di questa utility è DiskSalv.

Il programma è stato concepito per il recupero dei dischetti irrimediabilmente compromessi, ogni blocco del disco viene ispezionato per poi poter ricostruire le directory e i file.

Desiderate avere più spazio su un determinato disco, senza cancellare nulla? Detto, fatto! Il programma SQ vi può aiutare a diminuire le dimensioni di un file e addirittura a dimezzarle. Questa utility ci proponiamo di regalarvela nell'immediato futuro.

Se invece desiderate vedere come differiscono due file, c'è il programma DIFF. La sintassi per farlo funzionare è:

DIFF <file1> <file2>

L'elenco di programmi di questo tipo è veramente lungo, e noi preferiamo fermarci qui, se non altro per non farvi venire l'acquolina. Infatti, ci sembra quantomeno sadico parlarvi di programmi che forse non possedete e illustrarvene i dettagli. Vi annunciamo comunque che i programmi per la manipolazione dei file e altre utility stanno entrando di prepotenza in Disco Magazine per cui non disperate, presto sarete accontentati.

Come perdere la luna e vivere felici

QUAZIONI DIFFERENZIALI

di Luigi Manzo e Giovanni Michelon

In questo articolo vedremo alcuni metodi di risoluzione di equazioni differenziali; si tratta di un argomento di grande interesse pratico, oltre che teorico, perché la simulazione di un qualunque sistema fisico richiede la soluzione di insiemi più o meno grandi di tali equazioni.

Una certa conoscenza del problema e dei modi migliori per affrontarlo ci aiuterà dunque a realizzare con una certa consapevolezza la stesura sia di programmi "seri" riguardanti le più svariate discipline (dalla fisica all'astronomia, all'economia, alle scienze demografiche ecc.) che giochi di movimento (simulatori di volo, per esempio, o ovunque occorra rendere realistico il movimento di un oggetto).

Abbiamo deciso di spezzare in due puntate l'argomento: in questa prima parte verranno esposti alcuni fondamenti teorici ed i primi metodi di integrazione cosiddetti "ad un passo", sufficienti a comprendere lo schema di flusso del programma "Stars" presentato sul disco. Nella seconda puntata affronteremo i metodi "multipasso" ed alcune questioni relative all'errore ed alla stabilità dei vari metodi.

L'argomento "Equazioni differenziali" richiede purtroppo una corposa introduzione matematica e la definizione di alcuni concetti non proprio ovvi dell'analisi differenziale. Non vogliatecene perciò se anche stavolta vi tedieremo con un lungo preambolo teorico; i più ferrati in materia potranno comunque saltarlo, per arrivare direttamente ai metodi di integrazione numerica.

Per rendere più digeribile a tutti il concetto di derivata di una funzione, abbiamo pensato di introdurlo legandolo strettamente ad esempi tratti dalla cinematica, in modo da renderlo più familiare ed intuitivo. Si richiede comunque a monte la conoscenza del concetto di funzione, delle rappresentazioni nel piano cartesiano, ed un po' di familiarità con i vettori.

Tutto chiaro? Bene, possiamo incominciare.

Derivata di funzione

Storicamente, il concetto di derivata di una funzione è nato dell'esigenza di dare una soluzione ai problemi di meccanica riguardanti il moto dei corpi: anche noi, come già detto, cercheremo di introdurre tale concetto utilizzando esempi riguardanti tale disciplina.

Consideriamo quindi un corpo in movimento lungo una certa traiettoria; potrebbe trattarsi ad esempio di un'automobile in moto lungo una strada. Supponiamo di conoscere la sua "legge di moto": ossia supponiamo di conoscere in qualche modo quella funzione che ci dice la posizione dell'automobile in corrispondenza ad ogni istante di tempo che ci piaccia considerare. Questa legge ci può essere nota, in modo molto banale, attraverso un elenco: all'istante iniziale l'automobile si trova al punto di partenza, dopo un secondo è a dodici metri da esso, dopo due secondi a trenta metri, ecc. È chiaro però che un tale elenco non può essere esaustivo. Se volessimo conoscere la posizione dell'auto dopo 1.5 secondi, potremmo darne solo una valutazione approssimativa.

È molto più comoda in realtà un'espressione analitica di tale funzione: ossia una relazione matematica che ci permette di calcolare la posizione facendo opportune operazioni sull'istante di tempo che si va a considerare. Un esempio di legge del moto data in forma analitica può essere:

x(t) = 3*t*t

In base ad essa possiamo affermare che dopo un secondo l'auto si trova a tre metri dalla partenza, dopo 2 secondi a 12, dopo 3 secondi a 27 metri, ecc. Ovviamente, è piuttosto improbabile che gli sbalzi di umore del pilota dell'auto, da cui dipende la pressione del suo piede sul pedale dell'acceleratore e quindi la posizione dell'auto ad ogni istante, si lascino intrappolare da una formulazione così semplice. Tuttavia un numero consistente di importanti fenomeni di moto che avvengono in natura si lasciano schematizzare da leggi di questo tipo, e in ogni caso ci interessa mettere in luce la possibilità concettuale della loro esistenza.

Bene, andiamo avanti. Tutti sappiamo che cosa sia la velocità media di un corpo in moto: se il nostro mobile ha percorso 143 metri in 10 secondi, diremo che la sua velocità media è di 14.3 metri al secondo (ossia di 51.5 km/h). Tuttavia sappiamo anche che, istante per istante, la velocità

indicata dal tachimetro dell'auto varia e può essere anche molto diversa dal valore precedentemente calcolato: cerchiamo ora di definire meglio questo concetto di velocità istantanea.

Supponiamo di disporre della formulazione analitica della legge del moto, e di voler calcolare la velocità istantanea del mobile ad un certo istante t1. Un'idea potrebbe essere la seguente: consideriamo un istante di tempo t2 successivo a t1, calcoliamo la velocità media relativa a tale intervallo, quindi prendiamo un t3 più vicino a t1 di t2, calcoliamo ancora la velocità media, iteriamo tale procedimento con istanti di tempo sempre più vicini al nostro t1.

Otterremo una successione di velocità medie tutte diverse tra loro in generale, ma intuitivamente ci aspettiamo che tali valori differiscano sempre meno tra loro e tendano ad un certo valore, che assumiamo come velocità istantanea del mobile all'istante t1. In termini matematici più generali diremo che tale valore è la derivata prima rispetto al tempo della funzione che esprime la legge del moto, calcolata in t1. Per una trattazione più formale e per i simboli usati comunemente si veda la figura 1.0.

A questo punto si impongono due osservazioni:

1) Per un operatore "umano", calcolare la derivata di una funzione in base a tale procedimento, è un affare piuttosto serio: tuttavia l'analisi matematica ha escogitato tutta una serie di teoremi e trucchetti vari per eseguire tale operazione senza dover trovare materialmente la successione. Ossia, data una funzione della variabile t, f(t), è possibile trovare (sotto opportune condizioni di regolarità per la f che qui non consideriamo per semplicità) un'altra funzione f'(t) che ad ogni istante t associa il valore della derivata prima della f nel punto t.

2) Per un calcolatore, invece, tale estrapolazione non presenta alcuna difficoltà, almeno concettualmente. Tuttavia possiamo già prevedere che quando i valori in gioco diventano molto piccoli, inferiori all' epsilon di macchina, gli errori di calcolo rendono inutilizzabili i risultati ottenuti; dovremo perciò accontentarci di risultati approssimati. Queste conclusioni, per analoghi motivi saranno valide anche per i metodi di risoluzione delle equazioni differenziali, come vedremo.

A questo punto possiamo facilmente introdurre il concetto di accelerazione: definiremo accelerazione media il rapporto tra la variazione di velocità e l'intervallo di tempo in cui essa avviene. L'accelerazione istantanea è definita in modo del tutto analogo alla

0) Consideriamo una funzione y=f(t) definita in un intervallo $[t_i, t_f]$
Sia t_1 tale che $t_i < t_1 < t_f$, e t_2 tale che $t_i < t_1 < t_2 < t_f$.
Definiamo rapporto incrementale della funzione f relativa al punto t1 e all' incremento $t_2 - t_1 = \Delta t$ il rapporto:

$$\frac{y(t_2) - y(t_1)}{t_2 - t_1} = \frac{\Delta y}{\Delta t}$$

tra incremento della variabile indipendente e incremento della variabile dipendente.
Considerando rapporti incrementali relativi a Δt sempre più piccoli, osserviamo che il loro valore si stabilizza attorno ad un numero che definiamo derivata prima della y(t) nel punto t_1, ed indichiamo con la scrittura:

$$y'(t_1) \quad \text{oppure} \quad \left.\frac{dy}{dt}\right|_{t_1}$$

Fig. 1

velocità, operando però stavolta sulla funzione velocità anziché sulla legge del moto. In altre parole diremo che l'accelerazione è la derivata prima della velocità rispetto al tempo; nota che sia le legge del moto potremo ricavare la funzione velocità derivando una prima volta, e la funzione accelerazione derivando una seconda.

Concetto di equazione differenziale

Bene, direte voi, ma che c'entra tutto ciò con il nostro problema di partenza, le equazioni differenziali? C'entra eccome; bisogna sapere infatti che un'equazione differenziale non è altro che un'equazione che lega tra loro una funzione incognita ed alcune sue derivate successive. La soluzione di tale equazione è appunto questa funzione incognita.

In altre parole un'eq. diff. può essere vista come un problema che suona ad esempio così: "Cercasi funzione f(t) tale che la sua derivata prima moltiplicata per 3 e sottratta alla funzione stessa mi dia la sua derivata seconda".

L'espressione analitica di questo problema particolare è riportata in figura 2.0.

L'equazione differenziale che fa la parte del leone in tutta la meccanica classica ha una formulazione estremamente semplice ed è nota come seconda legge della dinamica (o legge di Newton): è riportata in figura 2.1.

Tale equazione, valida istante per istante e che si può ritenere ricavata dall'esperienza, afferma semplicemente che la derivata seconda della legge del moto moltiplicata per la massa, deve dare una certa funzione nota che dipende dal sistema studiato.

Osserviamo inoltre che tale legge esprime un legame tra vettori; la soluzione è in realtà una funzione vettoriale, una legge cioè che ad ogni istante di tempo associa un vettore che mi dà la posizione del corpo mobile nello spazio rispetto ad un certo sistema di riferimento. Noi abbiamo finora parlato solo di equazioni scalari; però le considerazioni fatte prima sono valide se pensiamo ai vettori come scomponibili nelle loro componenti rispetto al sistema di riferimento; in tal caso la legge di Newton equivale alle tre equazioni differenziali (se siamo nello spazio tridimensionale) scalari riportate in figura 2.2.

L'esempio riportato in figura 3.0 vuole chiarire tutto ciò: consideriamo il caso semplicissimo di un proiettile lanciato da un cannone e soggetto solo alla sua forza peso, supposta costante e rivolta verticalmente verso il basso; supponiamo cioè trascurabile l'attrito con l'aria.

Le equazioni differenziali cui si arriva sono riportate in figura 3.1; il problema è

0) Esempio di equazione differenziale:

$$f(t) - 3 * f'(t) = f''(t)$$

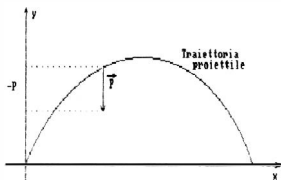
1) Equazione di Newton:

$$\vec{F} = \vec{m} * \vec{a}(t)$$

2) Equazione di Newton scritta nelle tre componenti scalari:

$$\begin{cases} F = m * a\ (t) \\ F = m * a\ (t) \\ F = m * a\ (t) \end{cases}$$

Fig. 2

B) Caso del proiettile puntiforme di massa costante m lanciato nel vuoto, sotto l'azione del suo peso P supposto costante.
Possiamo considerare un sistema di riferimento a due, anziche' tre, dimensioni perche gia' sappiamo che il movimento si svolge in un piano.

Valgono le seguenti condizioni:

1)
$$\begin{cases} 0 = m * x''(t) \\ -P = m * y''(t) \end{cases}$$

Osserviamo che:
- $x(t)$ e $y(t)$ sono le coordinate del proiettile all'istante t, e $x''(t)$, $y''(t)$ sono le derivate seconde di tali funzioni, ovverossia le componenti dell'accelerazione del proiettile secondo i due assi considerati.

- la componente del peso secondo l'asse x e' nulla perche' esso e' diretto verticalmente, mentre secondo l'asse y vale -P poiche' la forza peso e' diretta in verso discorde a tale asse.

2) Esempi di possibili soluzioni delle equazioni 3.1:

$$\begin{cases} x(t) = 3*t \\ y(t) = \frac{P}{2*m} *t^2 + 5*t \end{cases} \qquad \begin{cases} x(t) = 2 \\ y(t) = \frac{P}{2*m} *t^2 + 12 \end{cases}$$

3) Assegnate le seguenti condizioni iniziali:
v_{x_0} = componente sull'asse x della velocita' iniziale
v_{y_0} = componente sull'asse y della velocita' iniziale
x_0 = ascissa posizione iniziale
y_0 = ordinata posizione iniziale

l'unica soluzione dell'equazione 3.1 e':

$$\begin{cases} x(t)=v_{x_0}*t + x_0 \\ y(t)=\frac{P}{2*m} *t^2 v_{y_0}*t + y_0 \end{cases}$$

Fig. 3

ora risolvere tali equazioni, cioè trovare delle funzioni del tempo x(t) e y(t) che le soddisfano.

Chi conosce le regole di derivazione delle funzioni potrà constatare che esistono infinite soluzioni possibili; alcuni esempi sono riportati in figura 3.2. Ciò è del resto intuitivo; il movimento del proiettile dipenderà dall'alzo del cannone, dalla velocità che questo gli imprime inizialmente, e dal punto in cui si trova il proiettile al momento dello sparo. Solo specificando queste informazioni possiamo sperare di ottenere una soluzione ben determinata.

Infatti se specifichiamo le componenti della velocità iniziale del proiettile (in questo modo diamo l'informazione relativa sia all'alzo del cannone, sia alla velocità iniziale) e la posizione del proiettile all'istante dello sparo, l'equazione differenziale ha un unica soluzione riportata in figura 3.3.

In generale è possibile dimostrare che, detto n l'ordine di derivazione più grande che compare in un'equazione differenziale scalare (n si chiama semplicemente ordine dell'equazione), sono necessarie esattamente n condizioni iniziali per avere un'unica soluzione. Nel nostro esempio, avevamo due equazioni differenziali del secondo ordine, e sono state necessarie quattro condizioni (le due componenti della velocità e le due coordinate della posizione iniziale) per avere una ben determinata soluzione.

A questo punto sorge legittimo un dubbio: che utilità può avere un calcolatore in questo tipo di problemi se poi è possibile ricavare la soluzione espressa mediante una cosiddetta forma chiusa, ossia mediante una formula che mi permette di ricavare esattamente la posizione una volta assegnato l'istante di tempo?

Beh, in realtà ciò non è sempre possibile; anzi non lo è quasi mai nei casi di interesse pratico.

Tanto per fare un esempio, rendiamo un pochino più realistico il problema di prima tenendo condo dell'influenza dell'attrito dell'aria sul moto del proiettile. La fisica ci insegna che, se la velocità del corpo è sufficientemente elevata, l'azione dell'attrito dell'aria sul proiettile si può riassumere con una forza agente su di esso in verso opposto alla velocità, e la cui intensità è proporzionale al quadrato dell'intensità della velocità stessa; si veda la figura 4.0 per lo sviluppo analitico del problema, mentre nella 4.1 è riportato il sistema di equazioni differenziali cui si arriva.

Una semplice occhiata a queste ultime ci fa capire che trovare due funzioni che soddisfano a tali condizioni diventa decisamente complicato; quandanche ci si riuscisse, basterebbe alterare anche di poco la formula dell'espressione dell'attrito per dover daccapo cercare una soluzione completamente diversa dalla precedente. Se tutto ciò non bastasse, si consideri che per numerose, importanti classi di equazioni differenziali è stato dimostrato che non esiste una soluzione in forma chiusa esprimibile mediante le solite funzioni elementari. Per esse l'unica alternativa valida è rappresentata dall'utilizzo di un calcolatore.

Metodi approssimati

Storicamente, uno dei primi utilizzi seri del computer agli albori dell'epoca informatica è stato proprio la ricerca delle soluzioni approssimate di equazioni differenziali di grande interesse teorico e pratico (in termini espliciti: calcolo delle traiettorie delle bombe per i bombardieri americani nella seconda guerra mondiale); tuttora i mostri da svariati gigaflop al secondo dislocati in quei santuari dell'informatica pe-

sante che sono i centri di calcolo, passano gran parte del loro tempo ad integrare le equazioni differenziali che saltano fuori nei più disparati campi della fisica teorica ed applicata: dalla ricerca aerodinamica alla cromodinamica quantistica, dalla meccanica dei fluidi alla teoria dei campi ecc.

Noi, molto modestamente, prenderemo in esame una classe alquanto particolare di equazioni differenziali, e ci limiteremo a dimostrare alcuni tra i più noti metodi di integrazione numerica delle stesse; se siete interessati al calcolo delle traiettorie delle particelle di fluido in regime turbolento attorno al profilo alare di un Boeing 747 in assetto di volo, beh, ci dispiace, ma dovrete ricorrere a trattazioni un pochino più tecniche.

Precisamente, ci interesseremo di equazioni differenziali scrivibili nella forma:

$y'(t) = f(t,y(t))$

ossia di equazioni differenziali del primo ordine in cui sia possibile esplicitare la derivata prima della funzione incognita; il simbolo $f(t,y)$ indica una qualunque espressione nelle variabili y e t.

Queste restrizioni sono in realtà meno limitative di quanto possa sembrare; molte equazioni di ordine superiore al primo possono essere scritte come sistemi di equazioni di questo tipo e quindi sono risolubili con i metodi che vedremo.

Per garantire l'unicità della soluzione di un'equazione di questo tipo, è necessaria, come abbiamo visto una condizione del tipo:

$y(t0) = y0$

cioè di una relazione che ci dica quanto deve valere la soluzione in corrispondenza ad un certo punto t0. Al variare di tale condizione otterremo varie soluzioni; rappresentate su un diagramma, esse hanno l'aspetto di un fascio di curve non intersecantesi tra di loro. La condizione iniziale ci permette di selezionare una tra queste curve, e va sempre associata alla equazione differenziale considerata.

I metodi numerici di integrazione danno come risultato una successione di numeri, che indicheremo con y[k], che costituiscono una approssimazione più o meno accurata del valore della soluzione nel punto t[k]; t[k] è il valore della variabile indipendente dopo averla incrementata di k volte della quantità prefissata h. Possiamo scrivere:

t[k] = t[0] + k*h

Metodo di Eulero

È un po' il capostipite dei metodi di integrazione numerica, ed è il più semplice tra essi. Si basa sulla approssimazione della derivata prima della soluzione y(t) con il rapporto incrementale, utilizzando un incremento del tempo h sufficientemente piccolo.

Il ricavo matematico della relazione ricorsiva che permette il calcolo della successione è molto semplice ed è riportato in figura 5.0. Osserviamo che per il calcolo del punto (k + 1)-esimo è necessaria solo la conoscenza del punto k-esimo; il metodo di Eulero cioè appartiene alla famiglia dei metodi cosiddetti ad un passo, come il metodo di Heun ed i metodi di Runge-Kutta che vedremo più avanti.

Il metodo di Eulero si presta ad una semplice interpretazione geometrica che diventa chiara se si considera che la derivata prima di una funzione in un certo punto coincide con la pendenza della retta tangente al grafico della funzione nel punto considerato (vedi figura 5.1). Il grafico mette anche in evidenza l'errore compiuto utilizzando tale metodo, ossia la differenza tra il generico valore y[k] della soluzione numerica, ed il corrispondente valore y(t[k + 1]) della soluzione "esatta" dell'equazione.

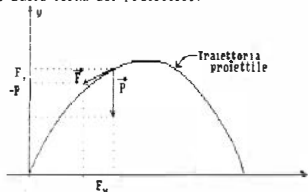
Metodo di Heun

Il metodo di Heun può essere inteso come un'evoluzione del metodo di Eulero:

8) Caso del proiettile puntiforme di massa m lanciato nell'aria sotto l'azione del proprio peso supposto costante.

La forza d'attrito e' espressa da:

$\vec{F} = -k*v^2*\vec{t}$ dove

- $\vec{t}$ e' il versore tangente alla traiettoria
- k e' una costante dipendente dalla viscosita' e dalla forma del proiettile.

Le componenti della $\vec{F}$ secondo gli assi della figura sono:

$$\begin{cases} F_x = -k*v*v_x \\ F_y = -k*v*v_y \end{cases}$$

La legge di Newton si scrive:

$\vec{P} + \vec{F} = m*\vec{a}(t)$

1) Le equazioni differenziali cui si arriva sono:

$$\begin{cases} -k*\sqrt{x'^2(t)+y'^2(t)}\ *x'(t)=m*x''(t) \\ -k*\sqrt{x'^2(t)+y'^2(t)}\ *y'(t)-P= m*y''(t) \end{cases}$$

Fig. 4

0) Calcolo della relazione ricorsiva di Eulero

Consideriamo l'approssimazione $y'(t) \simeq \frac{\Delta y}{\Delta t} = \frac{\Delta y}{h}$

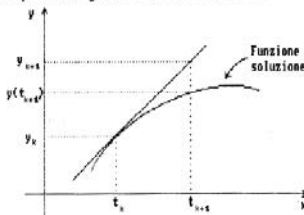
L'eq. diff. allora si scrive:

$$\frac{\Delta y}{h} = f(t,y(t)) \quad \text{Da cui:}$$

$$\Delta y = f(t,y(t)) * h \quad \text{oppure} \quad y(t+h)-y(t)=f(t,y(t)) * h$$

$$\text{o anche:} \quad y_{k+1}= y_k + f(t_k,y_k) * h$$

1) Interpretazione grafica del metodo di Eulero.

Fig. 5

più precisamente si consideri la retta passante per i punti (t[k],y(t[k])) e (t[k+1],y(t[k+1])) in figura 5.1. La conoscenza della pendenza di tale retta ci consentirebbe di ricavare il valore esatto della soluzione nel punto t[k+1], nota la y(t[k]). Disgraziatamente tale parametro non ci è noto; possiamo però pensare di ricavare un approssimazione migliore della semplice tangente alla curva per il punto t[k] (come fa in fondo il metodo di Eulero), considerando la media tra la pendenza della tangente alla soluzione in (t[k], y(t[k])) e la pendenza della tangente in (t[k+1],y(t[k+1])).

Tuttavia, al passo k-esimo non conosciamo il punto y(t[k+1]); allora ce ne procuriamo un'approssimazione utilizzando il metodo di Eulero. Da queste considerazioni nasce la formula riportata in figura 6.0; se la meditate un pochino dovreste riuscire a ripercorrere i passaggi visti sopra. In figura 6.1 è riportato un piccolo grafico ausiliario che illustra le varie notazioni.

È possibile dimostrare matematicamente che questo metodo presenta delle caratteristiche migliori rispetto al precedente; in termini un po' più tecnici si dice che esso ha un ordine di convergenza pari a 2, mentre Eulero ha ordine pari a 1. Ciò significa che per h sufficientemente piccolo il metodo di Heun dà in generale un'approssimazione migliore del metodo di Eulero.

Metodi di Runge-Kutta

A questo punto, può venirci l'idea di generalizzare ulteriormente la formula di Heun. Anziché considerare la media tra due pendenze, consideriamo una combinazione di più pendenze prese in punti intermedi tra t[k] e t[k+1]. Ciò che ne viene fuori è la formula, detta di Runge-Kutta, riportata in figura 7.0.

In essa n può valere 1,2,3 o 4 a seconda dell'ordine di convergenza che si vuole ottenere; le quantità k[i] che vi compaiono si calcolano con le formule di figura 7.1; osserviamo che per calcolare k[i+1] è necessario conoscere k[i].

Tutte le restanti quantità, cioè i coefficienti w[i], a[i], B[i,j], sono scelte in modo da rendere la convergenza migliore possibile. Per determinarle, occorre fare una lunga serie di calcoli, che noi, al solito, ometteremo. Ci limitiamo ad affermare che alla fine si trova che tali coefficienti devono soddisfare a sistemi di equazioni non lineari (diversi, a seconda del valore di n) aventi infinite soluzioni; si trova inoltre che se considerassimo formule di Runge-Kutta con n maggiore di 4, otteremmo sistemi senza alcuna soluzione. Si possono attuare alcune modifiche alla formula per riuscire ad ottenere ordini di convergenza più alti, ma i metodi che ne vengono fuori sono piuttosto pesanti da calcolare, e perciò sono poco usati; non li prenderemo pertanto in considerazione.

In figura 7.2 è riportato, come esempio, il sistema di equazioni cui si arriva con n eguale a due; osserviamo che una delle infinite scelte possibili per i coefficienti ci dà proprio il metodo di Heun. Nelle tabelle di figura 7.3 sono riportate alcune delle scelte più usuali; potete divertirvi a cercare altri coefficienti ed inventare così nuovi metodi di ordine due, ma possiamo assicurarvi che, a parità di ordine, tutti i metodi danno in generale gli stessi risultati.

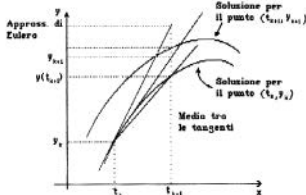



Fig. 6

Algoritmi e descrizione del programma

Vediamo ora come si utilizzano in pratica questi metodi; supponiamo di dover trovare la soluzione numerica di un equazione scritta nella solita forma : $y' = f(t,y(t))$, con la condizione iniziale $y(ti) = y0$ nell'intervallo [ti,tf]. L'algoritmo che realizza una generica formula di Runge-Kutta (e quindi, come casi particolari, i metodi di Eulero e Heun) è riportato in figura 8.0; come si può vedere, inizialmente si assegnano le condizioni iniziali alle variabili t[0] e y[0] e si azzera il contatore k dei passi eseguiti. Quindi si entra in un ciclo che incrementa k di un'unità e la variabile t[k] della quantita h, e da cui si esce quando t[k] è arrivata all'estremo destro dell'intervallo.

All'interno di questa struttura vengono calcolate le formule di figura 7; il ciclo su i serve a calcolare la formula 7.0, quello su j la sommatoria che compare nella formula 7.1.

Il risultato si trova memorizzato nelle tabelle unidimensionali t[k], y[k] (y[k] è il valore assunto dalla soluzione nel punto t[k]), le quali dovranno essere opportunamente dimensionate. Di solito è inutile memorizzare anche il valore della variabile indipendente, ma cio può essere utile in vista di successive elaborazioni del risultato.

A questo punto però possiamo essere comprensibilmente indotti ad un vago senso di frustrazione se pensiamo che anche la più banale equazione del moto è del secondo ordine e perciò non direttamente affrontabile con i metodi visti. Niente paura però: con un banale trucchetto possiamo comunque ricondurci al caso precedente. Consideriamo infatti un'equazione del secondo ordine scritta, o scrivibile, nella forma seguente:

$y''(t) = f(t,y(t),y'(t))$

Facciamo il seguente cambio di variabile: $v(t) = y'(t)$

L'equazione considerata è equivalente alle equazioni:

0) Formula di Runge-Kutta

$$y_{k+1} = y_k + \sum_{i=1}^{n} w_i * k_i$$

1) $$k_i = h * f(t_k + a_i * h, y_k + \sum_{j=1}^{i-1} b_{ij} * k_j)$$

2) $$\begin{cases} a_1 = 0 \\ w_1 + w_2 = 1 \\ w_2 * a_2 = 1/2 \\ w_2 * b_2 = 1/2 \end{cases}$$

3)

n ord. converg.	w_1	w_2	w_3	w_4	a_1	a_2	a_3	a_4	Metodo di
1	1				0				Eulero
2	1/2	1/2			0	1			Heun
2	0	1			0	1/2			R-K 2
3	1/6	4/6	1/6		0	1/2	1		R-K 3
4	1/6	1/3	1/3	1/6	0	1/2	1/2	1	R-K 4

Heun

$$B_{ij} = \begin{bmatrix} - & - \\ 1/2 & - \end{bmatrix}$$

R-K 2

$$B_{ij} = \begin{bmatrix} - & - \\ 1 & - \end{bmatrix}$$

R-K 3

$$B_{ij} = \begin{bmatrix} - & - & - \\ 1/2 & - & - \\ -1 & 2 & - \end{bmatrix}$$

R-K 4

$$B_{ij} = \begin{bmatrix} - & - & - & - \\ 1/2 & - & - & - \\ 0 & 1/2 & - & - \\ 0 & 0 & 1 & - \end{bmatrix}$$

Fig. 7

$y'(t) = v(t)$
$v'(t) = f(t,y(t),v(t))$
entrambe del primo ordine.

In figura 8.1 possiamo vedere l'algoritmo che porta alla risoluzione di questo sistema: non c'è alcuna novità di rilievo; osserviamo solo la duplicazione dei coefficienti K e delle variabili di appoggio.

A questo punto abbiamo tutti gli elementi per risolvere un problema di moto di un corpo puntiforme: ricapitolando, i passi da compiere, per il caso di un moto piano, sono i seguenti:

— si scrive la legge di Newton (fig. 2) con l'espressione della forza relativa al problema in esame

— si scrivono le equazioni valide per le componenti

— si fanno le posizioni adatte ad abbassare l'ordine delle equazioni pervenendo alla scrittura del sistema di fig. 8.2, del primo ordine.

— si risolve tale sistema con uno dei metodi visti.

Possiamo ora comprendere completamente le routines pubblicate in fondo all'articolo che costituiscono la parte centrale del programma relativo a questa rubrica, programma contenuto nel disco di questo mese.

Il programma si chiama "STARS" e simula l'evoluzione di un sistema di stelle (il cui movimento si svolge nel piano, per semplicità) sotto l'azione delle forze gravitazionali mutuamente attrattive. Esso è ispirato ad un articolo comparso sul numero 211 della rivista "Le Scienze", nella rubrica "Ricreazioni al calcolatore".

La struttura dati fondamentale che contiene le informazioni relative al sistema istante per istante è un array di record; la variabile S è di questo tipo. I dati relativi all'k-esima stella hanno perciò la seguente denominazione:

S[k].M = massa della stella
S[k].Vx = componente x della velocità della stella
S[k].Vy = componente y della velocità della stella
S[k].X = coordinata x della posizione della stella
S[k].Y = coordinata y della posizione della stella
S[k].Col = codice colore tracciamento della stella
S[k].Exist = flag di esitenza effettiva della stella

La variabile Sv è una copia esatta della S e serve a contenere i dati "vecchi" relativi al sistema prima di ogni integrazione.

L'accelerazione dovuta alla forza di mutua attrazione tra due masse è espressa, come noto, dalla legge di gravitazione universale riportata in figura 9.0; se introduciamo un sistema di riferimento le componenti di tale accelerazione sono riportate in figura 9.1. Ovviamente, per un sistema di n corpi, l'accelerazione subita dall'i-esimo è la somma delle accelerazioni dovute ai restanti n-1 corpi presi singolarmente. Notiamo inoltre esplicitamente che tale forza non dipende dal tempo, bensì solo dalla mutua posizione dei corpi; nelle equazioni differenziali perciò non comparirà esplicitamente la variabile tempo.

La procedura Acc ha in ingresso la variabile Sv che le fornisce la situazione del sistema prima di ogni integrazione, ed il record R che contiene la situazione dell'k-e-

```
0) Algoritmo di calcolo di una generica formula di Runge-Kutta di
   ordine n.

   h     ← valore del passo di integrazione
   t[0]  ← valore iniziale ascissa  ti
   y[0]  ← valore iniziale ordinata y0
   k     ← 0
   RIPETI
        t[k] ← ti + k*h
         yv ← y[k]
        PER i DA 1 A n
           z ← 0
           PER j DA 1 A i-1
              z ← z + Bij * K[i]
           FINE j
           valuta f(t[k]+ai*h,  yv+z)
         K[i] ← h* f(t[k]+ ai*h,  yv+z)
         y[k] ← y[k] + wi* K[i]
        FINE i
        incrementa k di 1
   FINCHE t[k] ≥ valore finale ascissa tf

1) Algoritmo di risoluzione di un sistema di 2 equazioni
    equivalente ad un equazione differenziale del secondo ordine:

   h        valore del passo di integrazione
   t[0] ← valore iniziale ascissa  ti
   y[0] ← valore iniziale ordinata y0
     V  ← valore iniziale per y'
     k  ← 0
     RIPETI
          t[k] ← ti + k*h
           yv  ← y[k]
          Vv   ← V
          PER i DA 1 A n
             zv ← 0
             zy ← 0
             PER j DA 1 A i-1
                 zv ← zv + Bij*Kv[j]
                 zy ← zy + Bij*Ky[j]
             FINE j
             Kv[i] ← h*f(t[k]+ai*h,v+zy,y[k]+zv)
             Ky[i] ← h*(Vv+zy)
             y[k]  ← y[k] + wi* Ky[i]
             V     ← Vv + wi* Kv[i]
          FINE i
          incrementa k di 1
     FINCHE t[k] ≥ valore finale ascissa tf

2) Sistema di equazioni del primo ordine per un generico
   problema di moto piano
```

$$
\begin{cases}
x'(t) = v_x(t) \\
y'(t) = v_y(t) \\
v_x(t) = \dfrac{F_x(t, v_x, v_y, x, y)}{m} \\
v_y(t) = \dfrac{F_y(t, v_x, v_y, x, y)}{m}
\end{cases}
$$

Fig. 8

simo corpo dopo un'integrazione "parziale". In uscita, la procedura fornisce le componenti Ax e Ay dell'accelerazione dell'i-esimo corpo: esse sono sfruttate nelle procedure IntegraHeun e IntegraEuleroche, sostanzialmente, implementano i metodi omonimi visti, relativamente all'i-esimo corpo. Per brevità sono riportati solo questi due metodi, ma voi stessi non dovreste avere difficoltà a scrivere le routines IntegraRK3 o IntegraRK4; esse si trovano comunque nei sorgenti riportati su disco.

È importante cogliere lo schema generale di calcolo del programma; esso si articola nelle seguenti fasi:

1) si fissa il corpo k-esimo

2) si calcola la sua accelerazione nella posizione iniziale e nelle varie posizioni intermedie richieste dal metodo

3) si integra per ottenere la posizione e la velocità

4) si memorizzano i risultati nella variabile S

5) si calcola l'accelerazione dell'(k+1)-esimo elemento considerando il corpo k-esimo nella posizione precedente all'integrazione.

Vi consigliamo, per comprendere meglio questo diagramma, di considerare inizialmente il metodo di Eulero, e solo più avanti i metodi di ordine superiore.

Infine, la procedura Integra: essa punta successivamente ai corpi del sistema, chiama le routine di integrazione, disegna la nuova posizione del corpo considerato, ed infine, dopo ogni passata su tutte le masse, aggiorna la situazione dell'intero sistema.

Utilizzo del programma

Il programma è stato compilato con il compilatore TDI-Modula 2 per Amiga, ver. 3.0. Per coloro che hanno una certa conoscenza di questo linguaggio, diremo che il programma è stato strutturato in due moduli separati, il primo dei quali, chiamato "StarUtils", contiene le procedure grafiche, relative al mouse, ai menù etc, mentre l'altro, il modulo principale "Stars", contiene le routines matematiche vere e proprie.

Il programma si apre con una schermata che mostra la regione del piano in cui avverrà l'evoluzione del sistema; a destra appare una scala graduata utile per settare la massa del corpo in esame; in basso a destra due gadgets con i quali si selezionerà la velocità di ogni stella.

Inizialmente bisogna fornire la disposizione e le velocità di partenza dei corpi del sistema; clickando col tasto destro del mouse, si deposita la stella nella regione voluta. Successivamente bisogna settare la velocità; ci si porta col mouse nella direzione voluta e si clicka: comparirà una freccetta che indica direzione e verso della velocità iniziale. A questo punto siamo liberi di stabilire il mòdulo della velocità, clickando sui tasti "+" e "-" ai lati della scritta "SPEED", e la massa operando, sempre col mouse, sulla scala graduata a destra. Si tenga presente comunque che le unità di misura sono arbitrarie: per costruire simulazioni realistiche sarà necessario costruirsi tabelle di conversione.

Se abbiamo sbagliato di individuare la posizione di una stella, possiamo clickare sul tasto "UNDO" per cancellare l'ultima posizione inserita.

Al termine dell'inserimento dei dati di una stella comunque bisogna clickare sul tasto "INSERT" ad indicare l'accettazione dei dati inseriti.

In alternativa a tutto ciò, possiamo caricare da disco dei file contenenti situazioni predefinite usando l'opzione "Load" del menù "Project" che viene evidenziato premendo il tasto destro del mouse. Ve ne sono un paio già definiti sul disco, relativi a situazioni piuttosto particolari. Potete goderveli direttamente.

Sempre con lo stesso menù poi è possibile decidere quale metodo di integrazione usare con l'opzione "Metodo".

Quando avrete terminato questa fase iniziale, premete il tasto "START" per dare il via al processo di integrazione. A questo punto non dovete fare altro che stare a guardare l'evoluzione del sistema; noterete che è abbastanza facile che due corpi vadano a scontrarsi (lo sarebbe molto meno in una più relistica simulazione tridimensionale); in tal caso si suppone l'urto anelastico, in modo che si venga a creare una nuova stella con massa pari alla somma delle precedenti, ed una velocità pari alla somma vettoriale delle velocità dei corpi all'istante dell'impatto.

Potete fermare ad ogni istante l'evoluzione del sistema clickando sul tasto "STOP"; ciò può servire per salvare su disco una situazione, per eventualmente riprenderla in un momento successivo; ciò si ottiene con l'opzione "Save" del menù.

Se invece vogliamo ridefinire daccapo una situazione, premiamo il tasto "INSERT" ; usando l'opzione "Clear" del menu inoltre, è possibile cancellare lo schermo. Tale operazione è consigliabile per evitare intrichi incomprensibili di traiettorie.

Il menù "Clear" è attivo anche durante la fase di immissione dati: in tal caso, esso è utile per azzerare completamente la situazione.

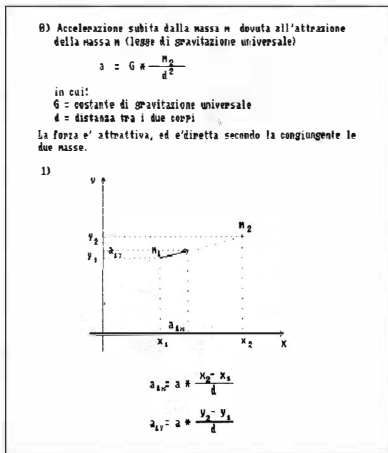



Fig. 9

LA TUA COLLANA PER CAPIRE, IDEARE, PROGETTARE

LIBRI DI BASE ELETTRONICA

ELETTRONICA IN AUTO

L'elettronica mantiene uno stretto rapporto con l'automobile, sin dall'istante della progettazione. In questa guida la presentazione dei circuiti elettronici in dotazione sulle vetture in circolazione e su quelle che lo saranno nel prossimo futuro. Un argomento di grande interesse e di estrema utilità.

AMPLIFICATORI OPERAZIONALI

Gli OP-AMP per la loro caratteristica di poter "eseguire operazioni", sono oggi impiegati e applicati in diversi settori dell'elettronica, nell'audio e alta fedeltà, nelle apparecchiature di laboratorio, nell'elettronica di commutazione. Questa guida ti permetterà di familiarizzare con questo componente elettronico moderno.

Se hai l'esigenza di conoscere per costruire tutto sull'elettronica, il Gruppo Editoriale Jackson ti propone i nuovi: "Libri di Base Elettronica", 20 preziose guide attraverso circuiti, componenti, grafici, fotografie e soprattutto innumerevoli idee per scatenare la tua fantasia con progetti collaudati e di immediata realizzazione.

VIDEOREGISTRATORI

In questo libro viene fornita un'idea di base sui differenti tipi di magnetoscopi esistenti in commercio, con particolare riguardo ai sistemi domestici da mezzo pollice e ai sistemi industriali U-Matic. Un intero capitolo è dedicato al montaggio, con consigli sulla manutenzione e sulla conservazione delle attrezzature.

LABORATORIO

Sei un appassionato di laboratorio, oppure miri a diventare un professionista del settore? Ecco un testo in grado di chiarirti le idee, verificare i tuoi progetti o insegnarti a regolare le apparecchiature già costruite. Un tester ed un saldatore per iniziare e poi la tua fantasia...

QUESTO MESE IN EDICOLA

"Fumetto in Computerart"
una produzione Graphic & Comp
Hardware: Amiga 2000
Polaroid Palette
Software: Deluxe Paint II

Felicita` e` un pasto caldo

Niente petrolio, ne` uranio...

testo
e disegni
di GOPI

e poi...mmm..

..c e`un panorama cosi monotono... ...uffa !

...che strazio...

mi viene una nostalgia di casa... ...SIGH !

Ma chi mai me lo ha fatto fare...

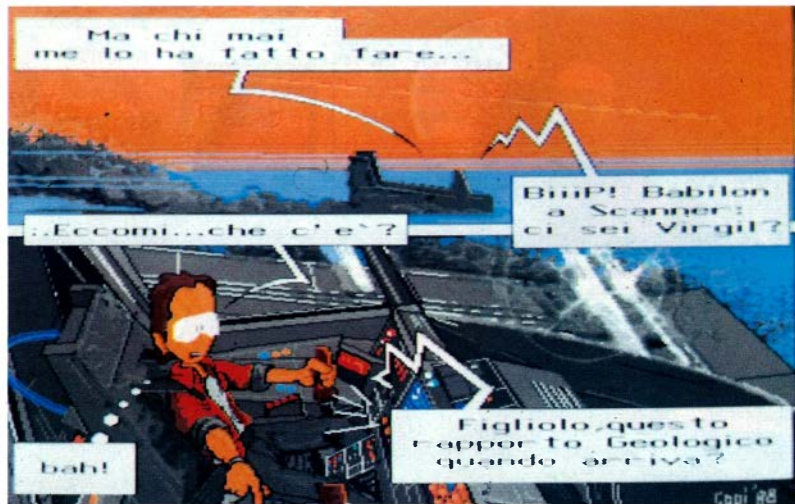
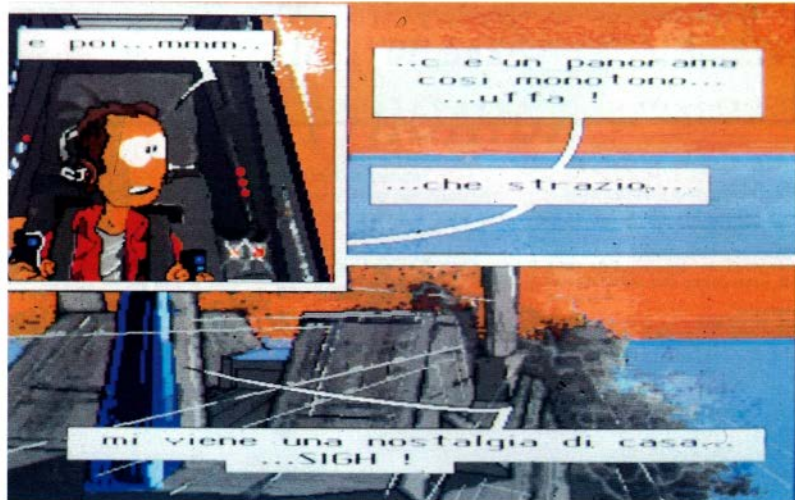
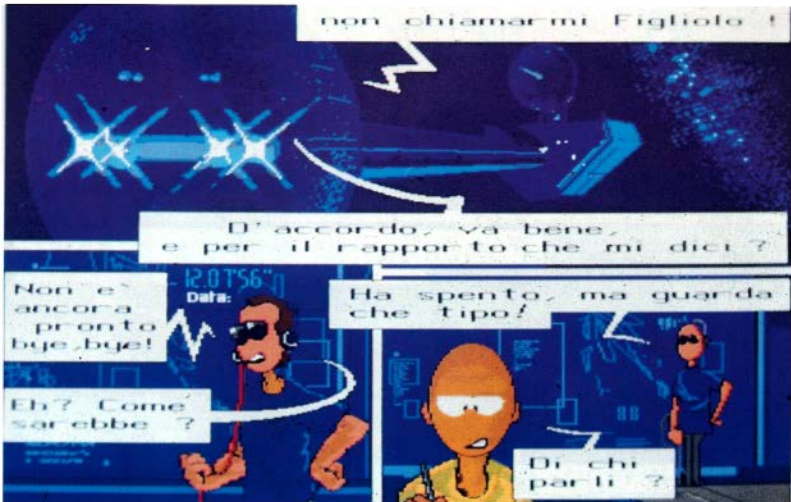
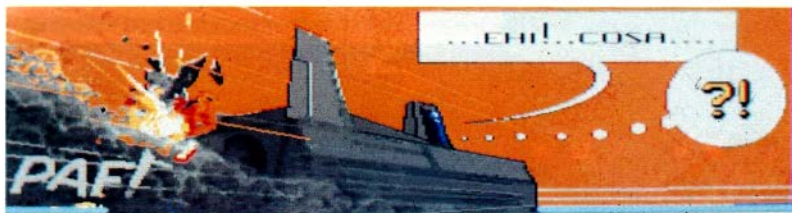
BiiP! Babilon a Scanner: ci sei Virgil?

:.Eccomi...che c'e`?

Figliolo,questo rapporto Geologico quando arriva?

bah!

Copi '88

Wow..! che disastro!
Bip-Bip! Rispondi "Scanner"
e adesso?
Rispondi Virgil...
Ho idea che la ricerca termini qui...
Accidenti !
Sara' meglio che avverta quei due in orbita che qui si e' rotto tutto!
Fzzz zzz
Una vera sfortu...
?
Bip! Rispondi Virgil...
continua...

ABBONAMENTO JACKSON = SERVIZIO COMPLETO

Da quest'anno l'abbonamento alle riviste Jackson offre una serie innegabile di vantaggi e servizi: anzitutto lo sconto eccezionale del 40% sul prezzo di copertina, pressochè doppio rispetto al passato, che Jackson ha voluto proporre ai lettori

SCONTO 40%

per celebrare il decimo anno di attività. Inoltre, abbonarsi a Jackson garantisce l'accesso a una rete multinazionale di informazioni, grazie al recente accordo azionario con la VNU Business Press Group, maggiore editore tecnico internazionale del settore. Ma c'è di più: **la Jackson Gold Card,** per l'identificazione immediata del codice abbonamento, sarà recapitata gratuitamente agli abbonati e permetterà al titolare di usufruire di molteplici servizi gratuiti quali: sconto del 20% fino al 28/2/1989 e del 10% dopo tale data, sul prezzo di copertina di libri e software Jackson, per acquisti effettuati direttamente dall'editore, oltre a una serie di sconti per acquisti vari presso librerie, computershop e altri esercizi convenzionati in tutta Italia.

In più, il titolare di **Jackson Gold Card** potrà ottenere sconti sui corsi di formazione della Jackson S.A.T.A., la scuola Jackson di Alte Tecnologie Applicate, oltre all'abbonamento gratuito a 6 numeri di uno (a scelta) dei tre settimanali Jackson: "E.O. News Settimanale di Elettronica", "Informatica Oggi Settimanale" o il nuovissimo "Meccanica Oggi", annunciato per l'inizio del 1989.

Infine, l'abbonato ha diritto all'invio personalizzato e riservato dei cataloghi libri e della nuova rivista **"Jackson Preview Magazine"**, con l'annuncio di tutte le novità editoriali Jackson.

1° PREMIO
HONG KONG • BANGKOK • SINGAPORE

2° PREMIO
UN COMPUTER AMIGA 2000

6°
UN COMPUTER

UN FANTASTICO VIAGGIO
IN ESTREMO ORIENTE
IN COLLABORAZIONE CON:

I LEADER PER UN VIAGGIO DI SUCCESSO

dal 7° al 10° PREMIO
4 PACCHETTI SOFTWARE
"Commodore Software by CTO"

REGOLAMENTO DEL CONCORSO

1 – Il Gruppo Editoriale Jackson S.p.A. promuove un concorso a premi in occasione della Campagna Abbonamenti 1988/1989.

2 – Per partecipare è sufficiente sottoscrivere, entro il 31.3.1989, un abbonamento a una delle 30 riviste Jackson.

3 – Sono previsti 10 favolosi premi da sorteggiare fra tutti gli abbonati.

4 – Primo premio: un viaggio per due persone in Estremo Oriente, che prevede: passaggi aerei Swissair, pernottamenti in Hong Kong, Bangkok e Singapore, presso gli hotel: Royal Orchid Sheraton e Sheraton Towers della catena Sheraton Hotel, nonché escursioni in luogo nelle tre suddette località.
Gli altri nove premi consistono rispettivamente (in ordine di esposizione) in:
1 computer Amiga 2000 completo di unità centrale con 1 MB di memoria, dischetto da 3" 1/2, tastiera, mouse, sistema operativo e monitor a colori 1084.
1 personal computer PC 20 III SERIE completo di unità centrale con 640 KB di memoria, dischetto da 5" 1/4, hard disk da 20 MB, mouse 1352, sistema operativo MS-DOS 3.20 monitor monocromatico e tastiera.
1 personal computer PC1 completo di unità centrale con memoria 512 KB, dischetto da 5" 1/4, tastiera, monitor monocromatico, sistema operativo MS-DOS 3.20 e GW-Basic.
1 computer Amiga 500 con 512 KB Ram e 256 KB Rom di memoria, sistema operativo e monitor a colori 1084.
1 computer "nuovo C64" completo di manuali e sistema operativo.
Dal settimo al decimo premio incluso, n. 4 pacchetti "Commodore Software by CTO".

5 – Gli abbonati a più di una rivista avranno diritto, per l'estrazione, all'inserimento del proprio nominativo tante volte quante sono le testate sottoscritte.

6 – L'estrazione dei 10 premi in palio avverrà presso la sede del Gruppo Editoriale Jackson entro il 30.5.1989.

7 – L'elenco dei vincitori, ad estrazione avvenuta, pubblicato su almeno 10 delle riviste Jackson. La vincita inoltre sarà pubblicata con lettera raccomandata a ciascuno dei sorteggiati.

8 – I premi verranno messi a disposizione degli aventi diritto entro 30 giorni dalla data dell'estrazione, ad esclusione del primo premio, il quale dovrà essere effettuato, compatibilmente con la disponibilità dei posti, in un periodo da definirsi, entro il 31.12.1989.

9 – Le spese di vitto relative al viaggio, nonché l'eventuale contratto di manutenzione extra garanzia per i personal computer Commodore, saranno a carico dei rispettivi vincitori.

10 – I dipendenti, i familiari, i collaboratori del Gruppo Editoriale Jackson sono esclusi dal concorso.

AUT. MIN. RICH.

ABBONAMENTO JACKSON = FORTUNA STREPITOSA

Abbonarsi alle riviste Jackson significa leggere il meglio, risparmiando il 40%, in informatica, elettronica e nuove tecnologie, ma soprattutto partecipare al grande concorso Jackson riservato agli abbonati, con la possibilità di vincere premi favolosi.

Sempre quest'anno, il concorso abbonamenti Jackson prevede un primo premio veramente eccezionale: la possibilità di esplorare il misterioso Estremo Oriente, in un viaggio che unisce il fascino di una tradizione millenaria ad uno sviluppo tecnologico senza precedenti.

Il viaggio, di oltre dieci giorni per due persone, è studiato nei minimi dettagli, per offrire il miglior comfort e le migliori ospitalità ed è garantito da tre leader di primissimo livello: Acentro Turismo di Milano, Swissair e Sheraton Hotels.

Non solo. Ad altri nove abbonati fortunati, il Gruppo Editoriale Jackson, in collaborazione con Commodore Computer e CTO, riserva altri premi eccezionali, dalla più completa gamma di computer di successo: un favoloso personal computer Amiga 2000, un Commodore PC20 III serie, un Commodore PC1, un Amiga 500 e un nuovo C64, in palio dal secondo al sesto estratto. Quattro pacchetti "Commodore Software by CTO" saranno inoltre sorteggiati dal settimo al decimo premio.

Partecipare al concorso è semplice: basta abbonarsi a una o più tra le riviste Jackson (chi si abbona a più riviste ha, naturalmente, più possibilità di vincita), utilizzando la speciale Cartolina/Questionario, già predisposta e affrancata, da compilare in ogni sua parte e restituire all'editore.

Affrettatevi! Abbonatevi per vincere!

GRUPPO EDITORIALE JACKSON
PRIMO NELLA BUSINESS-TO-BUSINESS COMMUNICATION

Abbonarsi è semplice: basta compilare in ogni sua voce la speciale Cartolina/Questionario già predisposta e affrancata e rispedirla all'editore.
Per il versamento dell'importo dell'abbonamento, utilizzate, preferibilmente l'apposito modulo di C.C.P. già predisposto e allegato alla rivista.

ABBONAMENTO JACKSON = RISPARMIO ECCEZIONALE

SERVIZIO QUALIFICAZIONE LETTORI

SPECIALE: PER CHI ACQUISTA LE RIVISTE JACKSON IN EDICOLA

Da quest'anno il Gruppo Editoriale Jackson ha predisposto uno **Speciale Servizio di Qualificazione Lettori e Abbonati,** che prevede l'assegnazione di una serie di dati relativi agli interessi specifici di ognuno, per poter offrire un servizio adeguato alle reali esperienze di aggiornamento del lettore.
Tutti i lettori interessati allo **Speciale Servizio Qualificazione Lettori,** e quindi anche i non abbonati, devono restituire, compilata nella parte **Qualificazione lettori,** la **Cartolina Questionario** già predisposta e affrancata.

Per chi la spedirà, il Gruppo Editoriale Jackson garantisce fin d'ora **GRATUITAMENTE:**

- Jackson Silver Card, che offre tutti i vantaggi della Gold Card, esclusi gli sconti sui libri riservati agli abbonati.
- Invio gratuito del **Catalogo Generale Libri Jackson.**
- Invio gratuito della **Jackson Preview Magazine.**
- **Abbonamento gratuito a sei numeri,** a scelta tra le seguenti riviste settimanali: E.O. News Settimanale - Informatica Oggi Settimanale - Meccanica Oggi (pubblicato da febbraio '89)

Area	Testate	Numeri Anno	Tariffa abbonam.	Tariffa intera
Elettronica e automazione	EO News Settimanale	40 + 6 omaggio	£. 59.500	£. 100.000
	Elettronica Oggi	20	£. 60.500	£. 100.000
	Automazione Oggi	20	£. 60.000	£. 100.000
	MeccanicaOggi	40 + 6 omaggio	£59.000	£.100.000
	Strumentazione e MisureOggi	11	£ 39.000	£. 66.000
Informatica e Personal Computer	Informatica Oggi Settimanale	40 + 6 omaggio	£. 61.000	£. 100.000
	Informatica Oggi mese	11	£. 33.500	£. 55.000
	BIT (quindicinale da Gennaio)	20	£. 48.000	£. 80.000
	PC Magazine	11	£. 32.500	£. 55.000
	PC Floppy	11	£. 79.500	£. 132.000
	Computergrafica e applicazioni	11	£. 39.500	£. 66.000
	Trasmissione dati e Telec.	11	£. 34.000	£. 55.000
	Compuscuola	10	£. 24.500	£. 40.000
Tecnologie e mercati	WATT(quindicinale da Gennaio)	20	£. 36.500	£. 60.000
	LAB. NEWS	10	£. 30.000	£. 50.000
	Industria Oggi	11	£. 34.500	£. 55.000
	Media Production	11	£. 46.500	£. 77.000
	Strumenti musicali	11	£. 32.000	£. 55.000
Hobby e Home Computer	Fare Elettronica	12	£. 36.000	£. 60.000
	Amiga Magazine disk	11	£. 92.500	£. 154.000
	Amiga Transactor	6	£. 25.500	£. 42.000
	Commodore Professional 64/128 disk	11	£. 85.000	£. 143.000
	CommodoreProfessional 64/128 cass.	11	£. 59.500	£. 99.000
	Supercommodore 64/128 disk	11	£. 79.000	£. 132.000
	Supercommodore 64/128 cassetta	11	£. 49.500	£. 82.500
	Olivetti Prodest User	6	£. 18.000	£. 30.000
	PC Software	11	£. 66.000	£. 110.000
	PC Games 5 1/4"	11	£. 93.000	£. 154.000
	PC Games 3 ½	11	£. 99.500	£. 165.000
	3 ½ Software	11	£. 99.000	£. 165.000

Lo sconto del 40% è stato calcolato, in certi casi, arrotondando le cifre in modo da differenziare le tariffe di ciascuna rivista per esigenze di gestione.

Il mondo sembrava rassegnato, all'improvviso una radiosa luce accecò gli umani, ancora una volta erano salvi...

di Alessandro Prandi

<<[...]Siete tutti così ridicolmente avvinti al vostro concetto di tempo-spazio! [...] È ora che la smettiate di pensare al tempo e allo spazio come se fossero inscindibili. Non lo sono affatto! In quanto tali neanche esistono! Non sono nient'altro che postulati della mente cosciente! Sono proprietà di corpi in movimento, concetti intellettuali necessari solo alla misurazione della fisica meccanica. Prendi la matematica: non conosce nè il tempo nè lo spazio. È così anche la fisica nucleare o lo spazio intergalattico. Nello spazio extraterrestre non c'è differenza alcuna tra un minuto secondo e un miliardo di anni! E allora perché cavolo dovrebbe esserci differenza nello spazio interiore che appartiene a ciascuno di noi? Nelle memorie di quel computer che chiamiamo mente sono immagazzinati milioni e milioni di anni! [...]>>.

Tratto dal romanzo "Stati di allucinazione" di P. Chayefsky pubblicato da Mondadori.

Disco Magazine ormai è decollato da qualche tempo e ormai per la terza volta sta bombardando il vostro Amiga con le nuove e affascinanti armi del sapere informatico. Nemmeno il famigerato scudo stellare, tanto agognato dalle superpotenze, potrebbe salvarvi dalla raffica di novità contenute in Disco Magazine III

Diffusa è in molti l'opinione che il sapere non paga, ma sicuramente, se avete seguito finora l'orbita del nostro Disco Volante vi sarete convinti del contrario e, quasi con matematica certezza, non mancherete ai prossimi appuntamenti con questo ghiotto boccone per il vostro computer.

Nei mesi estivi abbiamo sguinzagliato tutti i segugi a disposizione per una battuta di caccia un po' anomala ma alquanto redditizia. Diversi sono i carnieri pieni di nuovi programmi e curiosità, tutte per voi. Ovviamente possiamo offrirvi solo un disco al mese, per cui dovrete essere pazienti e accontentarvi di 880K alla volta. Fate finta di essere alle prese con un vino d'annata, e si sa che un buon vino và centellinato per poterne assaporare meglio il gusto e l'aroma.

Le vostre crescenti esigenze e la nostra innata curiosità ci hanno portato a scoprire un nuovo mondo disseminato di tesori e ricchezze, naturalmente già di pubblico dominio per quelli d'oltre oceano, ma ancora sconosciuto alla maggior parte di noi comuni mortali italiani. La ricerca è stata lunga e faticosa, ma ora possiamo rivolgerci a voi, sicuri di riuscire a suscitare interesse per ogni traccia del disco di Amiga Magazine.

Come al solito non tutte le ciambelle riescono con il buco e pertanto non accusateci se talvolta dedichiamo più spazio a questo o a quell'altro argomento, o programma, purtroppo accontentare tutti è pressocché impossibile. Non disperate prima o poi ognuno di voi troverà la sua ciambella preferita.

Menù di disk Magazine 3

Certamente, anche se non sempre realizzabile, riuscire ad accontentare le esigenze dell'utente medio è senza dubbio un obiettivo di primario interesse. Infatti, nella strutturazione del disco abbiamo tenuto conto dei relativi settori di interesse che possono risultare avvincenti anche per chi, di solito, rimane scettico e distaccato nei confronti del computer.

Per chi si avvicinasse per la prima volta a Disco Magazine va detto che il disco è suddiviso in Directory e Subdirectory in modo da facilitarne la lettura anche al nostro più sprovveduto Amigo.

Witz
Giochi
Musica
Grafica
Strumenti
Magazine

Il percorso disegnato all'interno del dischetto è abbastanza semplice, sia che vi si acceda da Workbench che da CLI troverete tutte le indicazioni necessarie per non perdervi. Un esempio di come è strutturato il disco lo troviamo proprio nella directory Witz. Accedendovi da Workbench troverete le icone dei programmi e una icona di una subdirectory (cassetto), ovviamente clickando su quest'ultima aprirete un altra finestra.

Per quanto riguarda la directory Witz ricordate che essa è l'indicatore del grado di umorismo esistente in redazione; non meravigliatevi dunque se talvolta percepirete una sensazione di eccessiva euforia: vorrà dire che qui tutto procede per il meglio. Per quanto possibile cercheremo sempre di allegare i listati sorgente alle divertenti gag contenute in questa sezione del disco.

Chi di voi vede nel gioco un indispensabile e irrunciabile componente del tempo speso davanti al monitor, non rimarrà deluso dai programmi contenuti appunto nella directory ad esso dedicata. In verità speravamo di potervi offrire già da questo mese un arcade di alto livello, ma purtroppo inconvenienti di ordine tecnico hanno fatto slittare questo appuntamento a data, comunque non lontana, da destinarsi.

Se per caso c'è qualcuno che ancora non è riuscito a sfondare nel mondo della musica o, che semplicemente sia ad essa appassionato, può sempre sbizzarrirsi con i programmi contenuti in questo anfratto di Disco Magazine, non sarà come vivere al Greenwich Village di New York, ma le potenzialità di Amiga potranno senz'altro ispirarvi qualche piacevole motivo. Se fate fortuna vi preghiamo di farcelo sapere, in fondo un po' di merito potrebbe essere anche nostro.

Se per caso vi imbattete nella directory Grafica vi consigliamo di esaminarla con attenzione, in essa infatti troverete sempre interessanti tool per il perfezionamento del disegno su Amiga. È inutile sprecare ancora parole sulle illimitate capacità di Amiga in questo campo, d'altronde, se l'avete comperato, di certo avrete già avuto modo di constatare la veridicità di tutte le lodi fatte a questo computer riguardo le illimitate potenzialità del suo chip grafico.

Per coloro che ormai possiedono l'Amiga da un po' di tempo e sono stanchi delle solite utility grafiche, musicali o, semplicemente, dei giochi e per chi comunque voglia conoscere la macchina in profondità, dedichiamo la directory Strumenti. Qui troverete i mezzi necessari per scoprire e capire i più nascosti chip del vostro imperscrutabile computer.

Per finire la descrizione sommaria del dischetto non manca che la directory Magazine che, come si può intuire dal nome, racchiude tutti i programmi sorgente contenuti nelle pagine della rivista. Ovviamente per qualsiasi delucidazione sul contenuto di questa directory dovrete prendere come punto di riferimento i vari articoli di Amiga Magazine. Per non tediarvi con un lungo lavoro di compilazione abbiamo preferito inserire qui anche i file eseguibili. Cos'è la gentilezza?

Un'ultima parola la spendiamo per le convenzioni adottate nella lettura dei file. Se noterete delle estensioni vicino ai nomi dei programmi sappiate che si tratta di:

.bas	sorgente in Basic
.c	sorgente in C
.h	sorgente header in C
.asm	sorgente in asm
.def	sorgente modulo definizione in Modula-2
.mod	sorgente modulo implementazione in Modula-2
.f	sorgente Forth
.scr	sorgente screen Forth
.o	file oggetto non linkato
.obj	file oggetto non linkato
.lsp	sorgente in Lisp

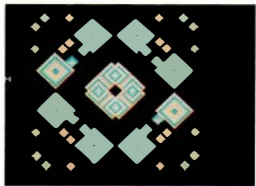
Ovviamente queste sono solo alcune delle estensioni, perlomeno quelle più frequenti, se, per caso nel disco, ne troverete altre diverse da queste appena elencate, sarà nostro compito delucidarvi in merito. Ricordate inoltre che alcuni programmi richiedono per il loro funzionamento l'area CHIP MEMORY, pertanto in quei casi, se il vostro computer è dotato di espansioni di memoria, dovrete selezionare l'icona NOFASTMEM nella directory SYSTEM del disco Workbench; una volta adempiuta tale formalità, quei particolari programmi funzioneranno in modo ineccepibile.

Witz

Sicuri di non essere stati scusati per aver mancato alla parola data, cerchiamo di farci perdonare. Nel numero 2 di Amiga Magazine avevamo accluso nel disco il programma Ombre, ricordate? Nella directory Witz per l'e-

sattezza. Purtroppo allora non fummo in grado di fornirvi il listato sorgente del programma. Vi preghiamo di voler essere indulgenti e seguirci nella directory Witz di questo mese e imboccare la subdirectory Ombre. Contenti? Si? Ora ci sentiamo più tranquilli e con la coscienza più leggera.

Per quanto riguarda il listato sorgente di Lente, sempre a riguardo di Disco Magazine II, dovrete attendere ancora un po', avremmo voluto inserirlo in questo disco, ma purtroppo lo spazio non è sufficiente.

La directory delle curiosità vi offre inoltre due programmi, Reticolo e Sfolgorio, che sicuramente attireranno la vostra attenzione. Entrambi sono programmi in C con allegati i rispettivi listati sorgente. Per quanto riguarda Sfolgorio vi consigliamo di non passare troppo tempo davanti al video mentre il programma sta girando, non vorremmo che rimaneste ipnotizzati dalle figure che in rapida successione appaiono in forma speculare. Fate quindi attenzione alle due finestre aperte da questi programmi.

<<Nel cielo, spinte da un vento irresistibile, correvano nere masse di vapori che, di quando in quando, lasciavano cadere sulle cupe foreste dell'isola improvvisi acquazzoni; sul mare, pure sollevato dal vento, s'urtavano e s'infrangevano furiosamente enormi ondate, confondendo i loro muggiti con gli scoppi secchi delle folgori.

Nè dalle capanne in fondo alla baia dell'isola, nè sulle fortificazioni che le difendevano, nè sui numerosi navigli ancorati al di là delle scogliere, si scorgeva alcun lume; ma in alto, sulla cima di un'altissima rupe tagliata a picco sul mare, brillavano due punti luminosi, due finestre vivamente illuminate>>.

E. SALGARI

Giochi

Nella directory Giochi potete scegliere fra ben tre diversi divertimenti. Niente di più ideale per ammazzare il tempo, che un solitario con le carte. Klondike è un gioco che vi farà passare qualche momento divertente, soprattutto in previsione delle buie e noiose giornate uggiose a cui andiamo incontro. Le carte sono quelle da Ramino, con l'unica differenza che il 10 è contradistinto dalla lettera T (ten). Queste vengono disposte in 7 colonne, in ordine decrescente da destra verso sinistra. Quattro caselle vuote vengono posizionate verticalmente nella parte sinistra dello schermo, sono riservate agli Assi. Il gioco consiste nel liberare le carte coperte nelle colonne alla destra e, allo stesso tempo, riporre con ordine, numero e seme, quelle disponibili nelle caselle vuote (A, 2, 3 eccetera). Per liberare le varie colonne dovete spostare per esempio il 5 sotto il 6, il J sotto la Q e così via. Regola inderogabile è che si devono disporre le carte a colori alterni (esempio: 5 di Cuori sotto il 6 di Picche). Se per caso avete la fortuna di liberare una delle colonne, sappiate che l'unico modo per rioccuparla è quello di depositarvi un Re (K). Se invece avete esaurito tutte le possibilità di abbinamento tra le carte potete riporre ancora qualche speranza nel mazzo coperto nella parte inferiore sinistra dello schermo. Cliccando in questo punto potrete disporre di nuove carte (compaiono nella parte superiore sinistra dello schermo) per proseguire il solitario. Il gioco finisce quando tutte le carte saranno riposte in ordine, seme e numero, nei 4 riquadri sopra menzionati.

Note: questo programma è stato scritto in ABasic. Se infatti entrate nella directory Klondike e prelevate il file con questo nome, lo trasferite in un altro dischetto e lo rinominate ABasic non otterrete altro che il vecchio e poco maneggevole interprete Basic della MetaComCo.

Passatempo è un gioco, in AmigaBasic, che ricorda molto ma molto lontanamente Space Invaders. Esso consiste nell'evitare al puntatore del mouse di cozzare contro le varie mattonelle che via via avanzano dalla parte inferiore dello schermo. Se date un'occhiata alle dimensioni del programma rimarrete sorpresi.

Terzo e ultimo gioco di questa volta è Cavallo. La dinamica del gioco evoca in modo inequivocabile gli scacchi. Dopo aver ascoltato la presentazione di Sir Archibald, avete di fronte a voi due opportunità: o premete la barra spaziatrice per ascoltare e vedere le istruzioni o premete <RETURN> per iniziare il gioco. Se optate per le istruzioni, sullo schermo compaiono le varie possibilità di movimento del Cavallo nel gioco degli scacchi. Capito il meccanismo o conoscendo già le regole scacchistiche, potete premere il tasto sinistro del mouse. A questo punto posizionatevi su una casella della scacchiera e ripremete il tasto del mouse. Ora il gioco ha inizio. Un contatore numererà le mosse che riuscirete a compiere, l'abilità sta nel non farsi sorprendere senza più alcuna possibilità di movimento a disposizione e, ovviamente, nell'occupare quante più caselle è possibile.

<<Nel principio degli anni, quando il mondo era ancora nuovo, nuovo, e gli animali cominciavano appena a lavorare per l'Uomo, vi era un Cammello, che viveva in mezzo a un gran deserto, perché non voleva lavorare; un Cammello che, straordinariamente pigro, mangiava piuttosto stecchi e spine e tamarischi e bacche ed erbacce; e quando qualcuno gli faceva qualche osservazione, rispondeva "Ob!" con disprezzo, che voleva dire "ohibò">>.

R. KIPLING

Musica

Per chi di voi non avesse ancora preso confidenza con la tastiera Amiga nell'esecuzione musicale, c'è un bel

regalo. Addentrandovi nella directory Musica, troverete un listato di 7 pagine in AmigaBasic che vi ispirerà nella composizione delle vostre melodie. L'uso del programma è quanto di più semplice esista. I menu selezionabili da mouse sono tre: Menu Forma d'Onda, Esecuzione Programma e Selezione Voce. Nel primo menu troviamo tre voci: Usa la F.O. Sin., Genera F.O., Usa la F.O. dell'organo. "Usa la F.O Sin." significa usa la forma d'onda sinusoidale; "Genera F.O." sta per costruzione della forma d'onda, in questo caso verrà aperta una finestra al centro dello schermo nella quale potrete disegnare la forma d'onda che più vi aggrada, provate ad usare più di 256 punti ma meno di 500. per uscire da questa finestra dovete posizionarvi con il puntatore del mouse appena un po' a sinistra della "F" di Forma d'Onda, all'esterno della title bar.

Per sentire le prime stridenti note dovete agire sui tasti della tastiera, la

Voce 1 corrisponde ai tasti 1, 2, 3... eccetera mentre la Voce 2 a quelli di A, S, D, F... eccetera. Per carità, non azionate le maiuscole!!.

<<Arrivarono presto alla casa, ch'era una casa di briganti. I briganti stavano intorno a una tavola lautamente imbandita. Allora gli animali tennero un consiglio e trovarono il modo per cacciar fuori i briganti. L'asino s'appoggio alla finestra, il cane gli saltò in groppa, il gatto s'arrampicò sul cane e il gallo volò sulla testa del gatto. A un dato segnale, poi, incominciarono tutti insieme il loro concerto. A quell'orrendo schiamazzo, i briganti atterriti, fuggirono nel bosco. I quattro compagni sedettero a tavola e mangiarono a più non posso>>.

J. e W. K. GRIMM

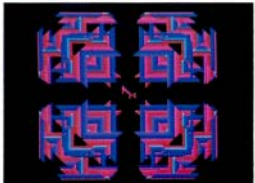
Grafica

Per incrementare le vostre fantasie cromatiche, abbiamo pensato di dotare questa directory di due tool grafici di estrema utilità per il perfezionamento dei vostri programmi. Infatti sia SpriteMaker che Palette vi possono tornare utili nella programmazione di tutti i giorni. Ricordate che il primo è in linguaggio C mentre il secondo è in AmigaBasic.

Il programma SpriteMaker vi permette di disegnare degli sprite e di convertire i vostri disegni in una struttura di data in C che può essere usata nei vari programmi. In questa procedura si può trovare una certa analogia con l'editor del puntatore che sitrova nel Preferences di Amiga.

Comunque il puntatore di default dell'intuition non viene modificato. L'immagine dello sprite generata dal programma può essere anche usata per gli VSprites.

Questa versione emette una struttura dati inizializzata, informazione colore, una routine in C per effettuare l'immagine dello sprite e il puntatore del-

l'Intiution per ogni particolare finestra aperta dal vostro programma. L'output può essere compilato come una subroutine separata e linkata con il vostro programma, o come una subroutine indipendente e quindi venire usata nelle più disparate applicazioni.

Per disegnare il vostro sprite, clickate il tasto sinistro del mause nel rettangolo grande nella parte sinistra dello schermo. Ogni pixel dello sprite apparirà come un piccolo quadratino. Una copia in dimensione reale dello sprite verrà visualizzata in un piccolo rettangolo nel mezzo dello schermo.

I colori del brush possono venire modificati selezionando i vari quadrati colorati nella parte inferiore destra dello schermo. Una volta scelto il colore da cambiare, basterà spostare gli indicatori di R, G e B lungo il rettangolo ad essi riservato.

Gli sprite possono arrivare sino a 200 pixel di altezza, ma non possono oltrepassare i 16 di larghezza. Lo sprite editor può essere spostato verso l'alto o in basso agendo sull'indicatore verticale al centro dello schermo.

Selezionado SAVE apparirà un requester che vi chiederà il nome da dare allo sprite e il nome del file da salvare nella struttura di data in C.

Clickando sul gadget POINT potrete determinare la posizione del punto principale dello sprite (come in Preferences), se non lo fate questo valore verrà attribuito per default e, corrisponderà all'angolo superiore sinistro dell'area dello sprite. Questa opzione presuppone che dobbiate usare lo sprite come alternativa al normale puntatore del mouse; se i propositi sono altri allora non dovrete selezionare questa funzione.

NOTA BENE:

L'Intuition causa la scomparsa dello sprite, se questo è usato come puntatore del mouse, ogniqualvolta il suo punto principale (PP) è settato oltre le 24 linee, partendo dall'alto dell'imma-

gine. Il PP sarà sempre attivo, ma l'immagine scomparirà come il puntatore verrà mosso verso la parte alta dello schermo.

Clickando sul gadget di RESET i colori dello sprite ritorneranno ai loro valori originali (di default) del programma.

Selezionando CLEAR IMAGE verrà cancellata ogni cosa dalla finestra editor.

QUIT/EXIT servono ad uscire dal programma ed a riportarvi al punto da cui siete partiti.

Notate anche che il colore dello sfondo (blu nei colori di default) non viene emesso poiché esso rappresenta il colore "trasparente" per lo sprite e, quindi, viene sempre ignorato. Esso viene usato per cancellare parti dello sprite nella finestra editor durante il disegno.

Il programma Palette invece, converte la Tonalità, la Saturazione e l'Intensità dei colori in valori RGB. D'ora

in poi niente più incertezze sui colori da inserire nei vostri programmi, ora avete un valido strumento per calcolare ogni più piccola sfumatura dell'immensa scala cromatica Amiga.

Strumenti

A questo punto del cammino all'interno di Disco Magazine, troviamo l'interessante reparto dedicato ai programmi di utilità. Il contenuto della directory Strumenti questo mese è particolarmente allettante. Oltre ad Agenda, utile programma per l'uso domestico, troviamo infatti un'indispensabile strumento per il lavoro quotidiano con il computer. Ma proseguiamo con ordine e compostezza!

Il programma Agenda viene presentato con l'intenzione di mantenere vivo in voi l'interesse per quel software dedicato all'uso frequente da parte dell'utente medio. Come avrete intuito si tratta di una agenda degli indirizzi Personalmente chi vi scrive, l'ha trovata molto utile nell'applicazione pratica. Come saprete, inevitabilmente dopo un po' di tempo, la classica agenda su carta risulta inservibile perché fisicamente logora ed alcune volte persino illegibile. A questo punto non rimane che munirsi di calamaio e piuma d'oca e ricopiare tutti gli indirizzi degli amici, e non, in un altra agenda destinata anch'essa alla stessa fine della prima, e così via.

Quale posto più sicuro e inespugnabile se non il vostro Amiga per gestire la cerchia delle vostre conoscenze? Di sicuro chi possiede un Modem capisce benissimo l'utilità di un agenda disponibile su disco, ma sicuramente anche gli altri non rimpiangeranno la vecchia rubrica telefonica della Banca tal dei tali o della Compagnia di Assicurazione pinco pallino.

Istruzioni per l'uso:

I comandi del programma possono essere inviati in gran parte indipendentemente, sia da tastiera che tramite mouse. I campi di scrittura possono ricevere sino ad 80 caratteri, se volete modificare questo valore, dovete agire sul programma sorgente. Ora vediamo le varie istruzioni.

Tasto Destro Mouse: sposta il puntatore nella parte bassa della finestra

RETURN: quando il puntatore è nella parte bassa della finestra entra nel modo edit

CTRL/RETURN: esce dal modo edit

Tasto Amiga D./Q: Restore del campo

TastoAmiga D./X: pulisce il campo

Cursore destro: record successivo

Cursore sinistro: record precedente

F1 - "FND": porta il cursore nel primo campo

F2- "SAV": salva i cambiamenti effettuati su un record esistente

F3- "ADD": aggiunge un nuovo record

F4 - "DEL": cancella un record

bfile.lib:

"bfile.lib" è un file che manipola il package scritto in C.

Un elenco sommario delle funzioni sopportate può essere il seguente:

```
carattere typedef            BFILE(100);
int esterno                  BFILE_OK;
BFILE FileData, FileIndice;
FileName (), RecordBuffer (), RecordKey ();
int RecordLength, RecordNumber, KeyLength,
Duplicates;
Crea un nuovo file dati
   MakeFile
(&DataFile,&Filename,RecordLength);
Apre un file dati esistente
   OpenFile
(&DataFile,&FileName,RecordLength);
Chiude un file dati
   CloseFile (&DataFile);
Aggiunge un nuovo record dati
   AddRec
(&DataFile,&RecordNumber,&RecordBuffer);
Cancella un record dati
   DeleteRec (&DataFile,RecordNumber);
Legge un record dati
   GetRec
(&DataFile,RecordNumber,&RecordBuffer);
Scrive un record dati
   PutRec
(&DataFile,RecordNumber,&RecordBuffer);
Ritorna il numero totale di record in un file
(anche se cancellati)
   FileRecs (&DataFile)
Ritorna il numero totale di record in uso in un
file
   UsedRecs (&DataFile);
Crea un nuovo file indice
   MakeIndex
(&IndexFile,&FileName,KeyLenght,Duplicates);
Apre un file indice esistente
   OpenIndex
(&IndexFile,&FileName,KeyLength,Duplicates);
   Chiude un file indice
CloseIndex (&IndexFile);
Aggiunge un key ad un file indice
   AddKey
(&IndexFile,RecordNumber,&RecordKey);
Cancella un key da un file indice
   DeleteKey
(&IndexFile,&RecordNumber,&RecordKey);
Resetta un file indice all'inizio
   StartKey (&IndexFile);
Ritorna il key e il numero di record che
corrispondono al key specificato
   FindKey
(&IndexFile,&RecordNumber,&RecordKey);
Ritorna il key e il numero di record che è
inferiore o uguale al key specificato
   SearchKey
(&IndexFile,&RecordNumber,&RecordKey);
Ritorna il prossimo key e il numero di record
   NextKey
(&IndexFile,&RecordNumber,&RecordKey);
Ritorna il precedente key e il numero di record
   PrevKey
(&IndexFile,&RecordNumber,&RecordKey);
(Molte funzioni settano BFILE_OK a VERO se
non ci sono errori e FALSO se ce ne stato uno)
```

Il programma DiskSalv ha come scopo, per quanto possibile, quello di recuperare completamente o parzialmente i dischi danneggiati. L'uso del programma è similare agli standard AmigaDOS usati per i vari DISKCOPY o COPY, il formato è:

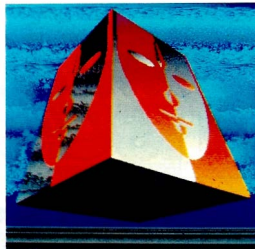
DISKSALV dfN: [TO] dfM:

Per disco sorgente s'intende quello danneggiato mentre il disco destinazione sarà un disco formattato, di solito, ma non necessariamente, vuoto. Il programma adopera due drive; non può recuperare df0: in df0: o in RAM:. Il salvataggio del disco danneggiato avviene oltrepassando la root-directory, la quale può essere rovinata (infatti è la causa più probabile per cui un intero disco viene rifiutato dall'AmigaDOS). Il programma effettua quindi una scansione per raccogliere tutte le directory e i loro relativi file, costruendo un'immagine dinamica interna del disco guasto. A questo punto

ogni directory trovata viene generata attraverso una normale chiamata DOS, di conseguenza ogni file relativo alle directory viene collocato al loro interno. Il procedimento continua finché non vengono più trovati File/Directory. Ricordate che non c'è alcuna relazione tra la locazione fisica dei file sul disco alterato e la locazione fisica dei file corrispondenti sul nuovo disco. Dopo che tutti i file linkabili sono stati ricollocati viene creata una directory di blocchi di dati esistenti ma inutilizzati, questo perché talvolta può essere possibile ricostruire manualmente l'header di un file, se andato distrutto. Il programma può anche copiare dischi buoni. Anche i file già cancellati verranno ricopiati, unica condizione perché ciò si verifichi è che non sia stato dato un SAVE sopra il file abolito. Questo aspetto del programma vi può certamente agevolare nel recupero di file erroneamente cancellati.

Questa versione di DiskSalv usa la sovvrabbondante informazione nell'AmigaDOS per scomporre ogni indirizzo d'aggancio file (file links) che risulta danneggiato. Ogni file nell'AmigaDOS ha un header contenente una lista di puntatori indirizzati ai vari blocchi (puntatori di blocchi) e un puntatore per il primo blocco. Questo primo puntatore è un link per primo blocco, 512 byte, di dati nel file; esso è anche ripetuto nella sopracitata lista di puntatori. Ogni blocco di dati possiede un link per il prossimo blocco che, a sua volta, può essere verificato per mezzo della lista dei puntatori di blocchi. I blocchi di dati contengono inoltre numeri in successione che indicano la sequenza logica nel file e dei puntatori di ritorno al blocco header del file. Questi due ultimi termini vengono controllati costantemente, tale procedura risulta estremamente utile per risolvere qualsiasi problema che possa sorgere nel linkaggio con il blocco successivo.

Questo programma è dotato inoltre di altri aspetti intelligenti. Esso segna ogni blocco alterato durante la scansione del disco. Poiché il DOS effettua diversi tentativi su ciascun blocco trovato, il provare a leggere un blocco danneggiato può comportare relativamente l'impiego di un lungo periodo di tempo, per cui marcando i blocchi da non leggere viene accelerato il processo di scansione. La seconda cosa che questo procedimento permette è quella di scrivere ogni specifico blocco dati. Se un blocco è già stato scritto una volta, esso viene segnato in modo da non venire usato di nuovo. Questo procedimento elimina tutti gli errori che potrebbero affiorare con una lista circolare di blocchi di dati, in special modo se un puntatore di blocco risultasse rovinato. Quando viene creata una nuova directory il programma verifica se già ne esiste una con lo stesso nome, se sì il nome verrà leggermente modificato. Dopo 10 tentativi il nome cambia. Il programma quindi passa alla scrittura della directory. Questo permette l'uso di un disco anche non vuoto per la ricezione dei file, senza doversi preoccupare di un'eventuale sovrapposizione delle directory sul disco di output.

Se per caso ad un certo punto della

fase di ricopiatura delle directory&file, il disco di destinazione risultasse insufficiente per contenere tutti i dati, il programma si ferma e vi chiede di inserire un altro disco per la ricezione di dati. La prima volta che il disco risulta pieno zeppo compare una finestra di request, clickando su CANCEL, una richiesta da CLI vi suggerirà di cambiare dischetto. Il programma proseguirà ricreando il path della directory sulla quale stava lavorando e copierà il file che non era riuscito a copiare nel disco precedente. Da questo punto in poi tutto procederà normalmente.

Un ultimo fatto di non trascurabile importanza, è che vi sarà possibile salvare i blocchi che hanno perso i loro file header. Di solito questi blocchi sono delle rimanenze dei vecchi file cancellati, ma potrebbero facilmente essere dei blocchi dimenticati da un normale file rovinato. Tutti questi avanzi vengono collocati nella directory SPARE-BLOCKS. Ogni file di questa directory avrà un nome che indica il numero di blocco al quale si riferisce, un numero blocco parent e un numero di sequenza. Ogni programma sorgente o altri testi ASCII possono essere messi insieme con facilità seguendo l'appropiata sequenza tra i file simili. Ogni file binario può essere analogamente riassemblato, ma solo quando tutti i blocchi nel file sono utilizzabili, il file riassemblato di solito risulta inutile.

Nota bene

La struttura del listato sorgente vi permetterà di sfruttare ogni piccola parte del programma per crearvi una serie di subroutine per i vostri lavori

<<Io non ho mai guadagnato nulla dal Sapere Supremo, e per questa ragione esso viene chiamato Sapere Supremo>>.

GOTAMA BUDDHA

Magazine

Anche questa volta siamo riusciti ad intravedere la via d'uscita di questo impossibile labirinto che si chiama Disco Magazine. Ma prima di terminare questo avventuroso viaggio vorremmo soffermarci nell'ultima camera, dedicata a Magazine. Se notate qualche programma di vostro gradimento tra le pagine della rivista, potete prendere armi e bagagli e trasferirvi in questa directory. Qui troverete sia i file sorgente che i programmi direttamente eseguibili, il listato della rivista diventa così solo un utile supporto per la consultazione tecnica. In questo modo speriamo di offrirvi un modo intelligente per approfondire i vari argomenti.

Il saluto

Ebbene è proprio giunto il tempo di lasciarci, pur sapendo di avervi dato abbastanza materiale per passare il tempo che ci separa da Disco Magazine IV, ci rimane sempre la sensazione di non aver fatto abbastanza per venire incontro alle vostre esigenze. Comunque, sensi di colpa a parte, vi auguriamo sinceramente di sollazzarvi il più possibile con ciò che vi abbiamo messo a disposizione.

<<Egli giaceva immobile con gli occhi chiusi, simile a chi dorme; pure ei non dormiva, giacché parlava con la sua anima. Ma l'aquila e il serpente, vistolo immobile così, rispettarono il silenzio e s'allontanarono cautamente>>

F. Nietzsche

Qualche opportuna presentazione preliminare

di Mr. Lambda

INGUAGGIO

In questa occasione attraverseremo nel modo più efficace, semplice e rapido possibile la sintassi del C, per presentarvi operatori e costrutti di controllo che il linguaggio vi mette a disposizione e il concetto di espressione, basilare per comprendere e utilizzare appropriatamente la potenza del linguaggio.

I ricordi relativi alle introduzioni oppure ai diversi tutorial per i linguaggi di programmazione richiamano in noi immagini con non poche frustazioni. Li ricordiamo sempre come troppo rapidi o troppo lenti. Se dipendesse solamente da noi, trascureremmo la parte introduttiva per arrivare subito a fare qualcosa d'interessante con il computer: pure lui non chiede di meglio!

Questo fenomeno può essere messo in relazione alla tendenza di assemblare qualcosa prima ancora di aver letto le istruzioni relative. Ci si rende conto di aver potuto trascurare qualcosa di critico e vitale, ma è più divertente e coinvolgente gettarvisi a capofitto. Quando poi le lezioni arrivano solamente una volta al mese, questa tendenza diventa irreversibile e irrimediabile.

All'arrembaggio.

Riusciamo ad immaginare diversi tipi di lettori di questi articoli sul C: il lettore curioso che vuol sapere qualcosa sul linguaggio C; il lettore curioso a tal punto da comprarsi pure un compilatore in C per Amiga, ma che non conosce il C; il lettore che spera di migliorare le sue conoscenze sul C; il serio programmatore in C che vuol vedere come vanno le cose; il lettore smaliziato e smaliziante, hacker potenziale, che spera di rintracciare qualche grossolanità per poter scrivere una lettera di commiserazione all'editore.

Naturalmente questo nostro incontro dovrebbe soddisfare solamente i primi tre gruppi di lettori, ma vedremo di fare qualcosa per soddisfare ogni palato. Innanzitutto questa volta ci cimenteremo con gli operatori del C, il concetto di espressioni in C, e gli statement di controllo in C. Speriamo anche di farci stare qualche utile ricetta. Anche il palato vuole la sua parte!

Chi ben comincia ...

Nell'articolo precedente, abbiamo descritto una semplice funzione chiamata 'mult()'. Essa ci ha fornito un buon approccio alla struttura fondamentale di qualsiasi funzione in C. La funzione ci ritornava un intero, e cioè il prodotto di due argomenti di tipo intero (integer). Ma vediamola:

```
/* una funzione per moltiplicare
   due numeri */

   int mult(a,b)
   int a,b;
   {
    int c; /* una variabile locale */
    c = a * b;
    return(c);
   }
```

Per richiamare la struttura base di qualsiasi funzione in un programma C, innanzitutto bisogna dichiarare il tipo di risultato fornito dalla funzione. Se questa è una funzione che moltiplica due interi, dovrebbe chiaramente restituirci un valore intero. Secondo, bisogna riportare il nome della funzione seguito immediatamente da una parentesi tonda aperta. Se alla funzione potranno essere trasferiti dei valori, è ne-

cessario elencare tutti i nomi locali di questi parametri, e quindi concluderli con una parentesi tonda chiusa. E subito dopo dichiarare il tipo di ciascun parametro in ordine. A questo punto si può cominciare a lavorare con la funzione stessa e cioè con la parentesi graffa d'apertura. Il corpo (body) di una funzione inizia infatti proprio con questa parentesi graffa aperta.

All'interno del corpo della funzione, le variabili locali possono essere dichiarate nel modo C standard: prima dichiarare il tipo della variabile — char, int, float, ecc. —, quindi il nome della variabile, concludendo la dichiarazione sempre con un punto e virgola (;).

Gli operatori

L'operatore di assegnamento del C '=' é fin troppo ovvio per qualsiasi abbia programmato in un linguaggio algebrico come il BASIC, Pascal, oppure FORTRAN. Ma il C ha una abbondanza di altri ope-

ratori tale da accontentare qualsiasi programmatore; alcuni di questi sono ovvi come quello che vi abbiamo or ora presentato, e altri sono molto più oscuri e richiedono una masticazione prolungata per essere digeriti.

Per l'addizione e la sottrazione con interi (integer) e in virgola mobile (floating point) il C ci mette a disposizione i tradizionali operatori ' + ' e '—', per la moltiplicazione ' * ' e per la divisione '/'. '%' è l'operatore del modulo, che potreste anche conoscere come operatore del resto. Come vi accorgerete presto, se non lo sapete già, il carattere '%' ha anche un altro significato per il compilatore, ma noi non ve lo anticipiamo, lasciandovi con un pizzico di suspance. E ora osserviamo una parte di codice sorgente in C che dimostra l'utilizzo di questi operatori:

```
a = 3;
b = 9;
d = a * b; /* d = 27 */
d = a / b; /* d = 0 */
d = b % a; /* d = 0 */
d = a << b; /* d = 1536 */
```

Semplice, no?!

Espressioni, espressioni ...

Un concetto chiave per comprendere i programmi in C è la nozione di espressione. Un espressione potrebbe essere per esempio un'espressione aritmetica, come quelle presentate sopra. Il C naturalmente dispone anche di espressioni logiche, con i corrispondenti operatori logici.

Un espressione può essere semplice come il nome di una variabile. Il valore della espressione 'Mele' è il valore memorizzato nella variabile 'Mele'. Il valore dell'espressione 'Mele + 10' è il valore memorizzato nella variabile 'Mele', più il valore 10. Questa espressione è di tipo 'int', se 'Mele' è di tipo 'int'. Il C possiede diverse regole per la valutazione delle espressioni. E quindi continuiamo la nostra discussione sulle espressioni.

Gli operatori logici

Gli operatori logici ci restituiscono valori che rappresentano il 'vero' (TRUE) e il 'falso' (FALSE), invece di un valore aritmetico.

Una espressione logica vera ritorna sempre il valore 1, mentre una espressione logica falsa ritorna sempre il valore 0.

E ora consideriamo alcuni operatori che vi dovrebbero essere familiari. '>' significa 'più grande di', '<' significa 'minore di', '<=' significa 'minore o uguale a', è '!=' significa 'non uguale a'. Mentre alcune versione del BASIC consentono l'utilizzazione di '=<', il C non lo permette. L'eguaglianza logica si esprime con '==', che in questo modo si distingue da '=', l'operatore di assegnamento.

Il '!' presente nell'operatore '!=', è pure lui un operatore logico, ma non relazionale come i precedenti. Esso viene chiamato 'not', e effettua il complemento logico dell'espressione logica: se l'espressione è vera, '!' la rende falsa. Ci sono poi altri operatori logici come '&&' per significare l'AND logico tra espressioni è '||' per significare l'OR logico.

Il C può anche manipolare facilmente i bit binari in una variabile. Gli operatori '<<' e '>>' muovono un valore a sinistra e a destra di determinate posizioni di bit. Uno shift sinistro viene completato con gli zero, mentre per lo shift destro le cose possono essere diverse. Ma non vi sembra strano che la definizione di un linguaggio di programmazione possa essere cosi imprecisa? Questo esempio evidenzia uno dei problemi che affliggono il C. Una versione del C può riempire lo shift destro con 0 e un'altra con 1. Spesso questo risultato dipende dal linguaggio macchina sottostante, cioè dal codice compreso dal processore del computer. Se il programmatore scrive un programma che dipende dai bit che devono venir riempiti da 1, il programma non lavorerà su una macchina che completa i valori shiftati con degli 0. Questo problema viene indicato con il termine di 'portabilità'. Queste differenze che possono sembrare irrilevanti tra le diverse implementazioni del C intralciano la trasportabilità dei programmi sorgente da un tipo di computer ad un altro.

Il C ha anche altri operatori per i bit binari. '&' effettua un AND, '|' effettua uno OR, '^' effettua uno XOR, e '~' un NOT.

Vediamo subito di dare qualche frammento di codice sorgente dimostrativo di qualcuna di queste operazioni:

```
int a, b, d;

a = 3;
b = 9;
d = a << b; /* d = 1536 */
d = a & b; /* d = 1 */
d = a | b; /* d = 11 */
```

Osservate che gli operatori logici ritornano effettivamente un valore aritmetico; 1 oppure 0. Questo valore può essere utilizzato in altre espressioni aritmetiche:

```
int a, b, d;

a = 3;
b = 9;
d = (a> b) + 1; /* d = 2 */
```

Le parentesi in 'd = (a > b) + 1' guidano il modo in cui il computer valuta un espressione. L'espressione all'interno della parentesi viene sempre valutata per prima, come in tutti i linguaggi algebrici. '(a > b)' è un'espressione logica che viene valutata come vera, in questo modo il valore aritmetico dell'espressione è 1, e la variabile d è il risultato di 1 più 1, cioè due. Inequivocabilmente conseguente, non è vero?

Per esempio, l'espressione aritmetica

```
(b + 3) * (d - 9)
```

è composta da diverse parti più piccole, '(b + 3)' e '(d - 9)', che vengono moltiplicate per assegnare all'espressione il valore di -92.

Le espressioni logiche possono presentarsi nello stesso modo delle espressioni aritmetiche, possono, cioè, essere composte di una serie di subespressioni. L'espressione logica

```
(a < b) > (b > d)
```

ha due parti , '(a< b)' e '(b > d)'. Entrambe devono essere dapprima valutate, e quindi i loro valori rispettivi vero o falso vengono confrontati con l'operatore 'più grande di'.

Espressioni, sempre espressioni

La nozione di espressione attraversa tutto il C. Le funzioni, per esempio, possono restituire valori che possono venire utilizzati in altre espressioni, e il valore dell'intera espressione può venire assegnato a una variabile, oppure utilizzato per modificare il corso del controllo del programma.

Strutture di controllo

La comprensione delle modalità e flusso del controllo in un programma in C è semplice se conoscete già il Pascal o il Modula-2, se voi siete invece un testardo programmatore in BASIC, le cose potrebbero complicarsi un poco.

If ... else

Il C possiede un insieme ricco di strutture di controllo. La più semplice è 'if':

```
if(a < b)
{
 d = a - b;
}
```

Tutti gli statement all'interno delle parentesi graffe verranno eseguiti se il valore in 'a' è minore del valore in 'b'.

Se solamente uno statement segue 'if', allora le parentesi graffe non sono necessarie, in questo modo:

```
if(a < b)
d = a - b;
```

ha lo stesso effetto del pezzo di codice riportato sopra.

Bene, e ora vediamo di complicare un poco le cose: 'if' può anche venire articolato ed esteso con 'else'. Così:

```
if(a < b)
{
 d = b - a;
}
else
{
 d = a - b;
}
```

Se 'b' è minore di 'a', allora verrà eseguito 'd = a - b'. Le parentesi graffe, come ormai dovreste già sapere, possono essere omesse, dal momento che ciascuna parte condizionale contiene solamente uno statement. Per cui si può riscrivere in questo modo:

```
if(a < b)
d = b - a;
else
d = a - b;
```

Questo fatto evidenzia molto bene quale sia il modo di considerare un insieme di statement in relazione a uno statement. L'idea che ne sta alla base è che qualsiasi insieme di statement racchiuso da parentesi graffe verrà eseguito come un singolo statement. Un diagramma sintattico sufficientemente generale per il costrutto 'if' può essere:

```
if(espressione logica)
statement oppure
{
gruppo di statement
}
```

E questo può facilmente venire esteso per includere 'else':

```
if(espressione logica)
statement oppure
{
gruppo di statement
}
else
statement oppure
{
gruppo di statement
}
```

Questo tipo di diagramma sintattico viene utilizzato per definire molte grammatiche di linguaggi per computer.

For

Il C ha pure un costrutto per il controllo chiamato 'for', simile al FOR del BASIC, del Pascal, e di molti altri linguaggi. Ma in C esso possiede una maggior flessibilità, come vi mostra questo esempio:

```
for(i = 0;i< 10;i = i + 1)
{
 d = a * b;
}
```

Lo statement all'interno delle parentesi verrà eseguito dieci volte, con la variabile 'i' che assume i valori da 0 a 9. Dopo la prima parentesi tonda, e prima del primo punto e virgola, voi potete inserire uno statement C che sarà eseguito una volta solamente, prima che il loop o ciclo inizi. Se voi desiderate incominciare il loop senza che nulla sia inizializzato, basta che scriviate il punto e virgola subito dopo la parentesi tonda aperta.

Lo statement di mezzo, e cioè quello tra i due punto e virgola, sarà eseguito prima di ciascuna iterazione. Se il suo valore è falso (zero), il ciclo terminerà. Notate che esso può essere una espressione logica, relazionale, oppure uno statement qualunque del C come 'a = b' Quando questa espressione viene valutata e il valore risultante posto in 'a', essa viene testata, e il loop potrebbe terminare basandosi su questo valore.

L'ultima espressione, prima della parentesi tonda chiusa, viene eseguita dopo aver eseguito gli statement del loop, e prima che il valore centrale sia testato nuovamente. Nell'ultimo esempio, la variabile 'i' viene incrementata di uno. Questa operazione è così comune nella programmazione che il C possiede operatori speciali per operare incrementi e decrementi. Grazie a ciò, lo statement 'i = i + 1' può essere scritto come 'i++', con quasi lo stesso effetto. E può anche essere scritto come '++i' All'interno delle espressioni o statement, ci sono due differenti tipi di effetti che può produrre questo operatore. Uno è il pre-incremento, indicato da '++i', che significa, 'incrementa la variabile i prima che sia valutata in questa espressione' L'altro, 'i++', significa 'incrementa la variabile i dopo che sia stata valutata in questa espressione' A questo punto il nostro loop 'for' potrebbe assumere quest'aspetto:

```
for(d = 0, i = 0;i< 10;i++)
{
 d = a * b;
}
```

Osservate i due statement che sono inseriti prima del primo punto e virgola. Questi due statement 'd = 0' e 'i = 0' sono separati dalla virgola, ed entrambi verranno eseguiti quando il loop inizia, e in questo modo le variabili 'd' e 'i' saranno inizializzate a zero. In questo caso, si sarebbe potuto utilizzare anche '++i' con lo stesso effetto. Gli statement che nell'esempio sono separati dalla virgola potrebbero essere parte di un statement 'if', in questo modo:

```
if(a < b)
d = 0, i = 0;
```

Essi verrebbero trattati come un singolo statement, ed entrambi verrebbero eseguiti se 'a' fosse minore di 'b' Naturalmente, separare questi statement e racchiuderli tra parentesi realizzerebbe un più chiaro stile di programmazione.

All'interno di un'espressione, gli operatori di decremento e incremento ci permettono forme ed effetti di programmazione interessanti e utili Per esempio:

```
int i,j;

i = 3;
j = 4;
d = i++ * ++j;
```

Significa che 'd' conterrà il valore 15, e 'i' e 'j' il valori 4 e 5, rispettivamente. Il '++j' significa che 'j' viene incrementato prima di venire valutato nell'espressione, in questo modo il risultato contenuto in 'd' è '3 * 5' oppure 15. 'i' viene incrementato dopo essere stato valutato. Ed eccovi un'altro loop 'for':

```
for(;i < 10; ++i)
{
 d = a * b;
}
```

In questo caso non succede nulla all'avvio del loop. Il loop poi verrà eseguito un certo numero di volte, in relazione al valore posseduto da 'i' prima dell'inizio del ciclo. Se 'i' è più grande di 9, gli statement del loop tra parentesi graffe non saranno eseguiti. E anche in questo caso, le parentesi graffe possono venire tralasciate se deve venire eseguito uno solo statement. Ma lo statement del loop potrebbe essere anche un'altro statement loop. Il costrutto in questione viene chiamato un loop 'for' nidificato:

```
for(i = 0;i < 10;i+ +)
  for(j = 0; j < 10; j+ +)
     d = d * i * j;
```

Questo tipo di struttura programmativa potrebbe essere utilizzata per inizializzare un array di valori per una costante, per esempio.

Questo tipo di nidificazione può verificarsi anche con il costrutto 'if':

```
if(a < b)
 if(c != d)
  g = 3;
 else
  g = 6;
```

La parte 'if(c != d) g = 3; else g = 6'; viene eseguita solamente se '(a<b)'. L''else' non si riferisce al primo 'if'. Ricordate che qualsiasi 'else' incontriate, dovete riferirlo all''if' più vicino.

Un'altra cosa importante in C, come in quasi tutti i linguaggi, è un coerente stile di indentazione. Naturalmente al compilatore non importa nulla del vostro stile, ma vi sembra facile districarsi se riprendeste in mano questo codice sorgente dopo qualche tempo, magari in qualche altro programma?!

```
 if(a < b)
if(c != d)
 g = 3;
 else
g = 6;
```

Ad una prima occhiata, sembra che l'autore del programma consideri lo statement 'g = 6' come clausola 'else' dello statement 'if(a < b)' Naturalmente il compilatore non può considerare le intenzioni del programmatore e collegherà l''else' a 'if(c != d)'

While, do ... while

'while' è un altro costrutto simile a 'for' Eccovi un esempio di un loop 'while':

```
while(a < b)
{
 b = b - 2;
}
```

Questo pezzo di codice potrebbe girare fino a che 'b' è minore di 'a'. L'espressione tra parentesi tonde viene valutata prima che gli statement tra parentesi graffe siano eseguiti, in questo modo, se l'espressione è falsa gli statement non saranno mai eseguiti. C'è pure una variante a 'while', chiamata 'do-while':

```
do
{
 b = b - 2;
} while(a < b);
```

Questa volta l'espressione tra parentesi tonde viene eseguita e testata alla fine del loop, dopo che gli statement nelle parentesi graffe sono stati eseguiti. Questo costrutto garantisce che gli statement tra parentesi graffe verranno eseguiti almeno una volta

Questo costrutto è simile al costrutto del Pascal 'repeat-until', eccetto che gli statement 'do-while' vengono eseguiti mentre l'espressione che rappresenta il test è vera, e gli statement 'repeat-until' del Pascal vengono eseguiti mentre l'espressione che rappresenta il test è falsa.

Qualche volta si utilizza il costrutto

```
while(1)
{
 ... il codice ..
}
```

Dal momento che l'espressione '1' sarà sempre vera, il loop sarà eseguito pre sempre, cioè è un loop infinito. Abitualmente, i loop di questo tipo includono uno statement 'break' all'interno:

```
while(1)
{
 ...
 ... if(error)
   break;
 ...
 ...
}
```

Il 'break' salta fuori dal loop, ed esegue il codice che segue la parentesi graffa chiusa del 'while'

In alcuni programmi possono accadere parecchie cose, in relazione allo stato di particolari valori. Questo può essere realizzato con diversi livelli di statement 'if':

```
if(a = = 1)
 b = 12;
else
 if(a = = 3)
   b = 8;
  else
   if(a = = 5)
     b = 4;
    else
     b = 2;
```

Questo costrutto viene abitualmente indentato solamente a un livello, per maggior chiarezza.

Switch

Lo stesso effetto può venire prodotto con un costrutto chiamato 'switch':

```
switch(a)
{
  case 1:
    b = 12;
    break;
  case 3:
    b = 8;
    break;
  case 5:
    b = 4;
    break;
default:
  b = 2;
}
```

Se 'a' non corrisponde a 1, 3, oppure 5, allora verrà eseguito il codice che segue il 'default:' Se il 'default:' e il suo codice viene rimosso, non viene eseguito nessun codice nel caso il valore di 'a' non corrisponda a 1, 3, oppure 5.

La parola chiave 'break' permette di saltare fuori dal controllo di 'switch'. Se il 'break' non termina ciascun 'case : b = xx;', allora il controllo del programma potrebbe passare attraverso i rimanenti gruppi di statement 'case', eseguendo di conseguenza gli statement 'b = 8', 'b = 4', e 'b = 2' Molto spesso lo statement break viene utilizzato prorpio in questo contesto.

Alla prova

Con questi operatori e queste strutture di controllo, più la semplice funzione di output 'printf()', voi potete prendere definitivamente confidenza con il linguaggio C. Uno dei modi migliori per imparare un linguaggio è 'ascoltarlo', leggendo e tentando di comprendere gli esempi di programmi C presenti per esempio nei manuali Amiga, nella nostra rivista e/o sul dischetto che l'accompagna, oppure ancora presenti in tutte quelle pubblicazioni che costituiscono la già immensa lettaratura di questo importante linguaggio. Non potete certo pretendere di comprendere tutto fin dall'inizio, ma partendo da ciò che vi è ormai familiare, potreste interpretare progressivamente le varie parti del codice sorgente fino ad averne una visione complessiva. La perseveranza e l'esercizio faranno il resto. Buon lavoro

Prossimamente

Nel nostro prossimo appuntamento parlermo del preprocessore, e impareremo a dichiarare altri tipi di variabili. Impareremo ad utilizzare diverse funzioni standard di input e output, insieme con la teoria Unix sugli argomenti della linea comandi. Alla prossima.

AMIGA, IN BASIC, OFFRE QUATTRO MODI PER INVIARE DATI ALLA STAMPANTE

E LE STAMPANTI

di Fabio Biancotto

Le due più importanti doti dell'Amiga sono la versalità e la facilità d'uso. Ma a volte queste qualità positive possono causare problemi agli utenti di Amiga che abbiano familiarità con altri sistemi di computer. Un primo esempio è il modo che l'Amiga usa per inviare i dati alle stampanti.

La maggior parte dei computer permettono di mandare i dati alle stampanti solo in pochi modi prescritti. La stampante è gestita per mezzo di uno o due DOS device possibili: la parallel device, se la stampante è attaccata al computer per mezzo della porta parallela Centronics, o la serial device se è attaccata alla porta RS-232. Per mandare dei caratteri alla stampante dobbiamo aprire il device che corrisponde alla porta a cui abbiamo attaccato la stampante, scrivere i caratteri che vogliamo mandare e in fine chiudere il device. I caratteri che vogliamo mandare direttamente alla stampante, non possono essere cambiati in nessun modo. Se vogliamo utilizzare delle funzioni speciali della nostra stampante, tipo stampa sottolineata o condensata, si dovrà inviare un codice specifico della stampante, direttamente alla stessa. Generalmente dobbiamo partire con un carattere Escape (in ASCII questo codice è il 27, o CHR$(27) in BASIC). Questi codici variano da stampante a stampante, pertanto il codice che vogliamo usare per chiamare una funzione speciale di una stampante non sarà utilizzabile su un'altra stampante.

Amiga offre invece quattro vie in basic

Funzione	Sequenza Escape*	Stampanti abilitate
Reset	c	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Inizializzato	#1	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Linefeed	D	Al Br D6 Cm Jx Ep Hp L+ Ql
Return + linefeed	E	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Reverse linefeed	M	Al Br D6 Jx Da Ql
Caratteri normali	[0m	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Italico on	[3m	Jx Ep Ok Hp L+
Italico off	[23m	Jx Ep Ok Hp L+
Underline on	[4m	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Underline off	[24m	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Boldface on	[1m	Al Br D6 Cm Jx Ep Da Hp L+ Ql
Boldface off	[22m	Al Br D6 Cm Jx Ep Da Hp L+ Ql
Setta colori foreground	[nm	Br Jx Da Dc Ql
(n sono due valori ASCII di numeri da 30-39)		
Setta colori background	[nm	Dc Ql
(n sono due valori ASCII di numeri da 40-49)		
Pitch normale	[0w	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Elite on	[2w	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Elite off	[1w	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Condensato fine on	[4w	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Condensato off	[3w	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Enlarged on	[6w	Cm Jx Ep Ok
Enlarged off	[5w	Cm Jx Ep Ok
Stampa ombra on	[6"z	Al Br D6 Cm Jx Ep Da Ql
Stampa ombra off	[5"z	Al Br D6 Cm Jx Ep Da Ql
Doppio strike on	[4"z	Al Br D6 Cm Jx Ep Da Hp L+ Ql
Doppio strike off	[3"z	Al Br D6 Cm Jx Ep Da Hp L+ Ql
NLQ on	[2"z	Cm Jx Ep
NLQ off	[1"z	Cm Jx Ep
Superscritto on	[2v	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Superscritto off	[1v	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Sottoscritto on	[4v	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Sottoscritto off	[3v	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Normalizza la linea	[0v	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Linea parzialmente sopra	L	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Linea parzialmente sotto	R	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Set caratteri US	(B	Cm Jx Ep Hp L+
Set caratteri Francesi	(R	Cm Jx Ep
Set caratteri Tedeschi	(K	Cm Jx Ep

Tabella 1: codici delle stampanti Amiga

da seguire per mandare i dati alla stampante. Il metodo più comune, impiega il device dell'AmigaDOS PRT:. Il device PRT: evita i due più grossi svantaggi presentati dal maneggiare l'output delle stampanti. Il primo, elimina il bisogno di specificare al programma se la nostra stampante è collegata alla porta seriale o alla parallela. Il secondo, la possibilità di attivare, dal programma, speciali comandi, come il sottolineato, senza specificare il codice usato dalla stampante che è collegata al computer.

Quando il programma prova ad accedere al device PRT: in un primo movento accadono due cose. La prima, il software che maneggia il device della stampante prova il device logico DEVS: per un file chiamato "System-Configuration" (il nome del device DEVS è automaticamente assegnato dalla directory "Devs" nella startup del disco). Questo file contiene i valori settati che abbiamo salvato dal programma Preference. Questi valori comu-

nicano al device PRT: se la nostra stampante è connessa alla porta parallela o seriale, quindi dove bisogna inviare l'output.

Il device PRT: legge anche la "System-Configuration" per determinare che tipo di stampante stiamo usando. Questo è determinato dal nome della stampante che abbiamo selezionato nella schermata "Change Printer" del Preferences. Il PRT: dopo aver letto il nome della stampante, prova a caricare un file "printer driver" dello stesso nome nella sub-directory Printers del device logico DEVS:. Se stampiamo la directory DEVS:Printers, potremmo vedere che contiene i file corrispondenti a ogni stampante listata nel Preferences. Questi "printer driver" comunicano al device PRT: esattamente quali sono i comandi disponibili della nostra stampante e come ci si accede. Contengono anche programmi che possono convertire le informazioni relative a delle schermate grafiche in dati di immagini grafiche per queste stampanti; sempre che naturalmente supportino delle stampe dot o color.

Una volta che il device PRT: ha letto il

Funzione	Sequenza Escape*	Stampanti abilitate
Set caratteri UK	(A	Cm Jx Ep Hp L+
Set caratteri Danesi I	(E	Cm Jx Ep
Set carattero Svedese	(H	Cm Jx Ep
Set caratteri Italiano	(Y	Cm Jx Ep
Set caratteri Spagnolo	(Z	Cm Jx Ep
Set caratteri Giapponese	(J	Cm Jx Ep Hp L+
Set caratteri Norvegese	(6	Cm Jx Ep
Set caratteri Danese II	(C	Cm Jx Ep
Proporzionale on	[2p	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Proporzionale off	[1p	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Proporzionale cancellato	[0p	Al Br D6 Da Ok Hp L+ Ql
Set proporzionale offset	[n E	Br Da Ql
Auto giustificazione sinistra	[5 F	Al D6 Jx Ep Da
Auto giustificazione destra	[7 F	Al Jx Ep
Auto giustificazione pieno	[6 F	Cm Jx Ep
Auto giustificazione off	[0 F	Al D6 Cm Jx Ep Da Hp
Spazio lettera (giustificazione)	[3 F	Jx Ep
Parola riempita (auto centraggio)	[1 F	D6 Jx Ep
Spaziatura linea 1/8"	[0z	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Spaziatura linea 1/6"	[1z	Al Br D6 Cm Jx Ep Da Ok Hp L+ Ql
Setta lunghezza forma n	[nt	Al Br D6 Cm Jx Ep Da Dc Ok Hp L+ Ql
Perf skip n (n>0)	[nq	Cm Jx Ep Da Ok Hp L+
Perf skip off	[0q	Cm Jx Ep Da Ok Hp L+
Set margine sinistro	#9	Al Br D6 Da Dc Ql
Set margine destro	#0	Al Br Da Dc Ql
Set margine top	#8	Al Br D6 Da
Set margine bottom	#2	Al Br D6 Da
T&B margini	[n1;n2r	Hp L+ Ql
L&R margini	[n1;n2s	Br D6 Cm Jx Ep Da Dc Hp L+
Cancella margini	#3	Br D6 Cm Jx Ep Da Dc Hp L+ Ql
Setta tab orizzontali	H	Br D6 Da Dc
Setta tab verticali	J	Br D6 Da
Cancella tab orizzontali	[0g	Br D6 Da Dc
Cancella tutti tab orizzontali	[3g	Br D6 Cm Jx Ep Da Dc
Cancella tab verticali	[1g	Br D6 Da Dc
Cancella tutti tab verticali	[4g	Br D6 Cm Jx Ep
Cancella tutti h & v tab	#4	Br D6 Cm Jx Ep Da Dc
Setta default tab	#5	Dr D6 Cm Jx Ep Da Dc
Extende comando	[n"x	

Tabella 2: codici delle stampanti Amiga

file printer driver della nostra stampante, è in grado di attivare i comandi speciali della stessa; per il programma, quindi, non è necessario mandare i codici specifici per la stampante selezionata per accedere ai comandi speciali. Invece, il programma può (e deve) mandare i codici generali Amiga per selezionare i comandi. Ci sono correntemente 75 codici di stampa supportati dall'Amiga. Una lista di questi codici appare in tabella 1.

Per esempio, per attivare la stampa sottolineata nella nostra stampante da un programma in Amiga BASIC, che potrebbe mandare il codice Amiga "Underline on" al PRT: (o LPT1: che è lo stesso in BASIC), incurante del codice "start underline" della nostra stampante, possiamo usare:

```
OPEN "PRT:" FOR OUTPUT AS #1
PRINT #1, CHR$(27) + "[4M", 'Codice
          Underline On'
CLOSE #1
```

Possiamo anche usare il comando BASIC LPRINT che completa la stessa cosa:

```
LPRINT CHR$(0) + CHR$(27) + "[4m";
```

Questo metodo elimina il problema di specificare l'OPEN e il CLOSE al device PRT:. Le linee di comando BASIC, LPRINT e LLIST mandano entrambe i loro output al device LPT1:BIN, che e' una variazione BASIC del PRT:. Comunque, si può aver notato che abbiamo preceduto il codice escape Amiga (CHR$(27) + "[4m") con un carattere non-stampato CHR$(0). Per molte ragioni, il device BASIC LPT1:BIN richiede che mandiamo un carattere non-stampato come CHR$(0) prima di mandare la sequenza escape. Se evitiamo di mandare il CHR$(0), il BASIC non passerà il codice escape Amiga corettamente alla stampante, e il comando speciale non sarà attivato.

Il vantaggio di usare il device PRT: al posto di una sequenza di escape alla stampante sarebbe ovvio. Un programma BASIC che usasse il PRT: per accedere ai comandi speciali di stampa per lavorare con qualche stampante potrebbe comunicare con il printer driver istallato, non con una sola particolare stampante. Quindi, per avere dei vantaggi dal device PRT:, dobbiamo aver istallato il printer driver della nostra stampante nel disco Workbench che usiamo come start up.

Ci sono due passi da seguire per istallare il driver della stampante sul nostro disco Workbench usato come start up. Il primo, dobbiamo settare il default della nostra stampante del programma Preferences e salvarlo sul disco. Il Preferences ci fornisce le informazioni in un file chiamato "System- Configuration" che è allocato nella directory "Devs" del nostro disco startup. Il secondo passo è quello di assicurarci che ci sia nella directory "Devs/Printers" il file driver della stampante che corrisponde a quella selezionata nel Preferences.

Il più importante default della stampante che dobbiamo settare dal Preferences è selezionare il tipo si stampante che vorremmo usare. Per specificare il nostro tipo di stampante, selezioniamo lo schermo "Change Printers" dallo schermo principale del Preferences. Una lista di stampanti è esposta nell'angolo in alto a destra dello schermo. Il nome della stampante, che è attualmente selezionato, è evidenziato. Per cambiare la stampante selezionata, basta cliccare in una delle due freccie per avere il prossimo nome nella lista. Un disco standard Workbench contiene un certo numero di stampanti differenti: la Alphacom Alphapro 101, Brother HR-15XI, Diablo 630, Commodore CBM MPS 1000, Epson X-80 (MX-80, FX-80, RX-80, LX- 80, eccetera), Diablo Advantage D25, Diablo C-150, Okidata Okimate 20, Hewlett Packard Laser Jet, Hewlett Packard Laser Jet Plus, la Qume LetterPro 20 ed altro ancora.

Se non ci fosse la nostra stampante nella lista, potremmo lavorare ugualmente scegliendo un altro default di stampante, questo perché molte stampanti usano gli stessi codici escape. Se possediamo una stampante a margherita possiamo selezionare dal Preferences una stampante, nella lista, a margherita come la Diablo 630 o la Qume LetterPro 20. Altre stampanti possono essere compatibili con i codici escape usati. Inoltre, fra le stampanti dot-matrix, la Epson ha selezionato uno standard simile per le proprie stampanti, includendo le adozioni dell'IBM. Per questa ragione, molte delle nuove stampanti dot-matrix sono Epson compatibili, tanto da includere altre note stampanti come la serie delle Star Micronics Gemini, la Citizen 120D e la Panasonic 1090 e le sue derivate. Potrebbe essere che molte stampanti che sembrano compatibili con le Epson non usino esattamente ogni singolo comando speciale di stampa, tanto per cambiare, non c'è solo uno standard Epson: il modello Epson MX-80 non ha gli stessi codici escape dei suoi modelli attuali. Per questa ragione il driver Epson non sopporta tutte i driver delle Epson-compatibili, particolarmente se la nostra stampante è di un paio di anni fa.

Se non possiamo lavorare con una di queste stampanti a disposizione nella lista, dovremmo istallare un driver di una stampante custom. Selezioniamo il "Custom" dalla lista delle stampanti e digitiamo il nome nel riquadro delle stampanti custom marcato "Custom Printer Name" che appare sotto la lista delle stampanti. Il nome di default è generico, installa un driver di stampante che ignora tutti i comandi speciali. Comunque può essere possibile ottenere un driver di una stampante custom che supporti completamente i comandi della nostra stampante.

Ci sono altre fonti da cui possiamo recuperare i driver. Una di queste è la stessa Commodore, attraverso la seconda relase del Workbench. La versione 1.2 sopporta le stampanti Apple Imagewriter II e la Okidata ML92. Ci sono molti altri programmi che provvedono a supportare le stampanti all'Amiga (come l'Okidata fatta per l'Okimate 20). Ci sono pure delle fonti commerciali per i driver delle stampanti: la Digital Creations vende driver per stampanti sia per la stampante a colori Apple Imagewriter II, come per le stampanti dot-ma-

trix a 24-pin Toshiba P1351, P351 e P351-C e la C.

Il disco di Public Domain Amicus numero 9 contiene i driver per le Cannon PJ 1080A e la C a getto di inchiostro, pure la Prowriter, la NEC 8025A, la Star Gemini 10, la Okidata ML92, la serie Panasonic KX- P109x, la Epson LQ800 e la Smith Corona D300. Altri driver potrebbero comparire in altri dischi. Naturalmente, se siete dei programmatori avventurosi (e abbastanza esperti), potreste scrivervi il vostro personale driver per stampante. Le istruzioni complete sono nel volume 2 del ROM Kernel Manual. Ma dovreste essere informati che non è un compito facile. Bisognerebbe creare parecchi file in assembler o in C e unirli insieme per produrre un driver da lavoro. Tuttavia si potrebbe aggiungere ai driver già esistenti comandi speciali che potrebbero essere utilizzati da altri programmi.

Dopo aver selezionato il driver della nostra stampante custom dal Preferences, non dimentichiamoci di copiare l'attuale driver della stampante nella directory corretta. Pertanto, il device PRT: per usare il driver, dobbiamo locarlo nella directory Devs/Printers del Workbench usato dal computer come disco start. Se possediamo solamente una stampante possiamo liberamente cancellare tutti gli altri driver nella directory Devs/Printers. Questo ci permetterà di aver maggior spazio nel nostro disco Workbench.

Supponiamo che il nome del volume contenente il driver sia DRIVER e il nome del volume del nostro Workbench sia WORKBENCH, possiamo trasferire il driver sul disco Workbench con il comando CLI:

```
COPY DRIVER:Nome-driver to
     WORKBENCH:Devs/Printers
```

Inoltre, bisogna assicurarsi che il nome che abbiamo inserito nel riquadro "Custom Printer Name" del Preferences sia esattamente lo stesso del nome del file del driver della stampante.

In aggiunta per selezionare dal Preferences il tipo di stampante, ci sono molti altri accorgimenti a cui dobbiamo fare attenzione. Il tipo di connessione di default per la stampante è la porta parallela, mentre se usiamo una stampante seriale la collegheremo alla porta modem, essendo sicuri di aver selezionato l'icona "Serial" nell'angolo sinistro in alto della schermata "Change Printers". Possiamo specificare la lunghezza della carta che stiamo usando; se usiamo i moduli continui oppure i fogli singoli. Possiamo selezionare la spaziatura di 6 o 8 linee per pollice e settare il margine destro e sinistro. Si può scegliere anche il tipo di carattere (Pica, Elite o Condensato) ed inoltre la stampa draft o near-letter.

Mentre tutti i programmi hanno accesso a queste opzioni del Preferences, questi sono liberi di adottarle o ignorarle. Molti programmi presentano dei loro menu permettendoci di selezionare queste opzioni e altre usate espressamente. Si ricordi che sebbene un programma abbia un menu per selezionare i margini, questo è comunque influenzato dalle selezioni del Preferences. Perfino alcuni programmi che stampano schermate grafiche possono essere influenzate dal valore dei margini. Il morale della storia è che se non possediamo un appropriato printout che aggiusti i valori settati dal programma dobbiamo provare a cambiare i valori settati dal Preferences.

L'ultima opzione selezionabile dal Preferences è lo schermo "Graphic Select". Per molte stampanti dot-matrix, i valori settati sono buoni e non bisogna cambiarne alcuno. In molti casi, abbiamo però bisogno di fare molti cambiamenti. Se abbiamo una stampante a colori, sicuramente selezioneremo il "Color" dal riquadro "Shade" nella parte bassa dello schermo. Di default le schermate grafiche sono stampante orrizontalmente, però per ottenere un'immagine più larga possiamo selezionare la stampa verticale. A volte si potrebbe trovare che una immagine stampata verticalmente non ha un aspetto proporzionale (immagini che sembrano schiacciate orrizontalmente possono essere stampate facilmente nella versione verticale). Mentre, se stiamo stampando molte immagini che comprendono una palette di colori chiari e uno sfondo scuro possiamo stampare l'immagine in negativo, in modo da farla risaltare rispetto allo sfondo.

Una delle cose da ricordare del programma Preferences, in fine, è la possibilità di operare il salvataggio delle opzioni selezionate su disco. Dopo aver salvato su un disco le opzioni, possiamo trasferirle in un altro disco semplicemente copiando il file "Devs/System-Configuration" dal disco alla directory "Devs" del nuovo disco.

Sebbene usare il device PRT: sia un modo semplice per la maggior parte delle applicazioni dirette alle comunicazioni con la stampante, toccheremo brevemente gli altri metodi che sono disponibili. In addizione al device PRT: ci sono altri due device AmigaDOS che possiamo usare per inviare i dati alle stampanti. Questi due sono il SER:, il device seriale, e il PAR:, il device parallelo. Per prima cosa, bisogna conoscere quale stampante è connessa all'Amiga, poi possiamo scegliere o la porta parallela o la seriale. Quindi possiamo aprire quel device e mandare i dati alla stampante. Possiamo fare questo direttamente dal programma, o usare uno dei comandi DOS come il COPY o il TYPE per mandare il contenuto del file al device. I caratteri che mandiamo a una delle due porte non sono esaminati o tradotti in ogni caso, per cui, per richiamare i comandi speciali, dobbiamo mandare l'esatto co-

Tutti i programmi originali della Collana Commodore Software by C.T.O. li troverete presso i Commodore Point.

Chiedi ogni mese al tuo Commodore Point le novità

mando di stampa necessario alla particolare stampante che stiamo usando. L'uso di PAR: deve essere ristretto a quelle situazioni dove possiamo essere sicuri del tipo di stampante usata; come per il software, che è utilizzabile solo su una macchina. Il PAR: può essere molto utile nei problemi di disturbo nel posizionamento di una stampante. Se non siamo certi se il problema che dobbiamo risolvere risiede nel software relativo al driver della stampante o nella connessione hardware alla stessa, possiamo mandare direttamente i data al PAR:. Poiché questi dati sono diretti all'output della stampante direttamente, senza nessuna conversione, si potrà verificare così la corretta ricezione della stampante.

Il metodo finale per accedere alla stampante con l'Amiga è purtroppo riservato ai soli programmatori. I programmatori, esperti che hanno dimestichezza a lavorare con i device hardware del sistema operativo, possono bypassare i device delle stampanti AmigaDOS e indirizzare i device hardware "printer.device". Questo device assegna al programmatore il pieno controllo di tutta la stampante. Permette di mandare sequenze di codici escape Amiga che saranno tradotte dal driver della stampante nei codici specifici della stampante o di mandare i caratteri direttamente alla stampante senza nessun processo. In ogni caso, il programmatore non ha bisogno di sapere di come sia collegata la stampante all'Amiga.

Il programma BASIC che segue dà esempi di codici escape mandati alla stampante attraverso il device PAR: e attraverso il device PRT:, ambedue aprono il device direttamente e usano il comando LPRINT:

```
***** '* * '* Questo programma dimostra le tre
vie per mandare i codici* '* escape alla nostra
stampante dall'AmigaBASIC. Il primo * '*
manda i codici specifici della stampante ai
device PAR: o* '* al SER: (dipende da come la
stampante è collegata * '* all'Amiga). Il
programma scritto userà questo metodo * '*
lavorerà correttamente con la stampante
specifica. Ovvio* '* che questo metodo non è
raccomandato a meno che non ci * '* sia
un'altra scelta. * '* * '* La sezione seguente
manda i codici escape al device PRT:. * '*
Programmi che usano questi metodi potranno
lavorare * '* correttamente con ogni stampante
correttamente * '* istallata dal Preferences. * '*
* *****

'Setta le stringhe usate dai codici escape della
stampante

A$ = CHR$(27) + "!" + "Stampa allargata
     on"
B$ = CHR$(27) + CHR$(34) + "Stampa
     allargata off"
C$ = CHR$(27) + "!" + "Sottolineato on"
D$ = CHR$(27) + "J" + " Sottolineato off"

     'Manda le stringhe al device PAR:

OPEN "PAR:" FOROUTPUT AD #1
PRINT #1, A$;" ";B$;CHR$(13)
PRINT #1, C$;D$;CHR$(13)
CLOSE #1

'Setta le stringhe usate dai codici escape di una
stampante generica Amiga

A$ = CHR$(27) + "[6w" + "Stampa allargata
     on"
B$ = CHR$(27) + "[5w" + "Stampa allargata
     off"
C$ = CHR$(27) + "[4m" + "Underline on"
D$ = CHR$(27)+ "[24m" + " Underline off"

     'Manda le stringhe al device PRT:
     (anche chiamato LPT1:
     dall'AmigaBASIC)

OPEN "PRT:" FOR OUTPUT AS #1
PRINT #1, A$;" ";B$
PRINT #1, C$;D$
CLOSE #1

'LPRINT manda l'output al PRT: (attualmente al
device BASIC LPT1:BIN)
'Per lavorare con i codici escape di Amiga
bisogna ricordare
'di mettere per prima un CHR$(0) o un altro
carattere non- stampato.

LPRINT CHR$(0)+A$;" ";B$
LPRINT CHR$(0)+C$;D$
END
```

Sebbene possiamo usare uno dei tre metodi mostrati per mandare i data alla stampante, dovremmo essere consapevoli che non sempre è possibile usarli tutti in un programma. Tutti i tre metodi usano la stessa porta harware della stampante, perciò, non è possibile aprire il device PAR: quando è già aperto il device PRT: senza generare un errore "file already open". Quando comunichiamo con il PAR: o il PRT:, è possibile chiudere un device prima di aprirne un altro. Usando LPRINT apriamo il device della stampante e lo lasciamo aperto fino a quando non è ultimata la stampa. L'unica via per chiudere la stampante prima del comando LPRINT è usare un CLEAR, che chiude tutti i file, però azzera tutti i valori delle variabili usate dal programma. Questo metodo non è molto pratico per provare ad aprire il device PAR: e PRT: nello stesso programma che usa il comando LPRINT.

Il problema finale, che molti possessori di Amiga hanno riscontrato usando le loro stampanti, è come fare una copia delle schermate grafiche sulla stampante; questo è comunemente chiamato uno screen-dump. Gli IBM e gli IBM-compatibili hanno una combinazione speciale di tasti che possono essere premuti per mandare le schermate grafiche alla stampante in ogni momento. L'Amiga, comunque, non ha un simile comando. Ci sono invece delle routine del sistema operativo che automaticamente compiono un screen-dump: però bisogna avere dei programmi che chiamino questa routine. Molti programmi, tipo il Notepad e Deluxe Paint, provvedono a delle stampe di schermate operando facilmente dalla drag bar. Ci sono anche un certo numero di programmi in commercio che ci permettono di mandare qualunque schermata grafica alla nostra stampante.

Note

Le abbreviazioni usate per le stampanti nella tabella sono:

Al	=	Alphacom Alphapro 101
Br	=	Brother HR-15XL
D6	=	Diablo 630
Cm	=	Commodore CBM MPS 1000
Jx	=	Epson JX-80
Ep	=	Epson X-80 serie
Da	=	Diablo Advantage D25
Dc	=	Diablo C-150
Ok	=	Okidata Okimate 20
Hp	=	Hewlett Packard Laser Jet
L+	=	Hewlett Packard Laser Jet Plus
Ql	=	Qume LetterPro 20

* Tutti i codici escape devono essere preceduti dal carattere ESC (ASCII 27 o CHR$(27) in Basic). In aggiunta, quando mandiamo i codici escape al device PRT: via comando LPRINT in AmigaBASIC, dobbiamo far precedere il carattere ESC da un carattere non-stampato, come un CHR$(0). Per esempio, per mandare il codice per il "Sottolineato ON", possiamo mandare il carattere 27 al device PRT:, seguito dalla stringa di caratteri "[4m". In BASIC possiamo racchiudere il tutto con il comando

```
LPRINT CHR$(0) + CHRS$(27) + "[4M"
```

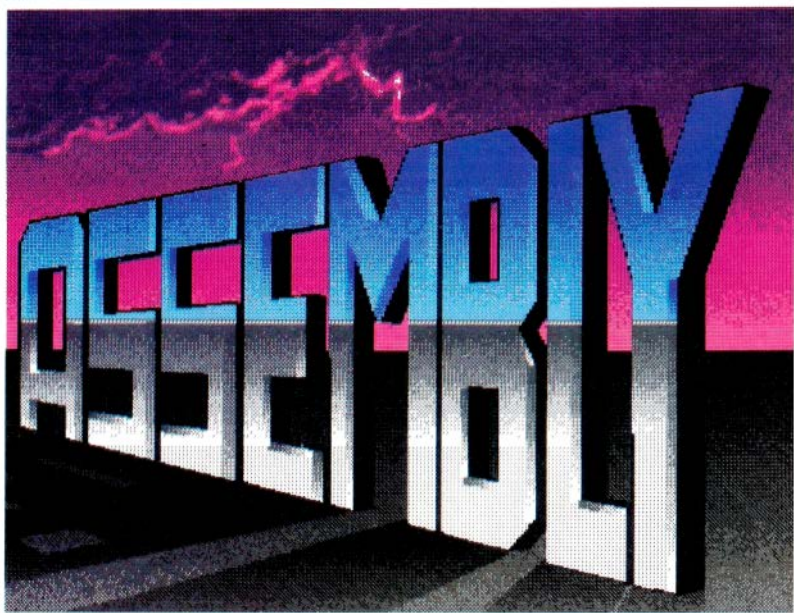
Nella tabella, la lettera n sta ad indicare il numero decimale, espresso come un set di ASCII digitati, per esempio "12". Per esempio, per selezionare la lunghezza di 50 linee, possiamo mandare il carattere ASCII 27 seguito dalla stringa "50t". In BASIC potrebbe essere letto:

```
LPRINT CHR$(0) + CHR$(27) + "50t"
```

Corso di

I flag e le istruzioni di controllo

TERZA PUNTATA

di Paolo Russo

La filosofia dei salti

In ogni linguaggio esiste una serie di istruzioni che vengono definite 'di controllo', il cui scopo consiste nel consentire iterazioni, diramazioni e chiamate a sottoprogrammi; si possono citare a esempio le parole chiave IF, THEN, GOTO, GOSUB, RETURN, FOR e NEXT tipiche del Basic classico. Nei linguaggi strutturati come il Pascal troviamo una serie di istruzioni di controllo che consentono soltanto iterazioni e diramazioni, oltre alla chiamata di sottoprogrammi, ma non in generale salti da una parte all'altra del programma; il GOTO talvolta esiste, ma il suo uso viene sconsigliato a tal punto che tale parola risulta del tutto assente nella maggior parte dei programmi, essendo stata messa al bando come una sorta di paria tra tutti i comandi esistenti nel linguaggio. In Assembly vige la filosofia esattamente opposta: a parte le subroutine, non esiste praticamente alcun modo per controllare un programma se non tramite salti espliciti. Ciò è essenzialmente dovuto a una esigenza di semplicità ed efficienza a livello hardware; il GOTO è infatti la più semplice e veloce direttiva di controllo immaginabile, tramite la quale si possono realizzare tutte le altre. In altri termini, tutti i compilatori esistenti al mondo traducono le strutture dei linguaggi evoluti in sequenze di GOTO quando devono generare il codice macchina; in teoria un ipotetico programmatore Assembly incapace di pensare se non in termini di tali strutture potrebbe eseguire mentalmente una procedura simile, ma in pratica quasi nessuno lo trova necessario.

Esistono varie forme di GOTO in Assembly, tra le quali si può scegliere di volta in volta la più adatta in base a diversi criteri, tra i quali innanzitutto il fatto che il salto debba avvenire sempre o soltanto sotto certe condizioni.

I salti incondizionati

La forma più semplice di salto è quella incondizionata: quando il processore incontra un'istruzione di questo tipo deve saltare in ogni caso. Il salto incondizionato di uso più diffuso è il BRA (da branch, ramo o diramazione) che deve essere seguito da una label. Sono possibili due varianti: la BRA.S che consente di saltare a una distanza massima di 128 byte e che occupa due byte di memoria di programma e la BRA semplice con la quale si possono raggiungere locazioni distanti 32K ma che richiede quattro byte. La S sta per Short, ossia forma corta. Simile discorso per l'istruzione BSR (Branch to SubRoutine) che, opzionalmente disponibile in variante BSR.S, è l'analogo Assembly del GOSUB. È sempre opportuno usare la forma corta del salto, quando possibile. Il difficile consiste nello stimare a occhio se un salto rientra nel limite di 128 byte o no; si può allora adottare la seguente tattica: quando si è indecisi si usa la forma corta e se il limite viene violato l'assembler lo segnala. Una strategia meno efficiente ma più sbrigativa consiste nell'usare la forma lunga in tutti i casi incerti.

Qualcuno sarà già inorridito: è mai possibile che il programmatore debba essere distratto da simili dettagli? Non può forse occuparsene l'assemblatore stesso? Purtroppo gli assembler più diffusi non sono in grado di scegliere autonomamente la variante giusta, per motivi tecnici che sarebbero in realtà superabili con un po' di buona volontà da parte delle software house produttrici di assemblatori. Molti assembler sono capaci di decidere automaticamente la lunghezza dei salti all'indietro ma non di quelli in avanti, che sono la maggioranza.

Non esiste GOSUB senza RETURN; l'istruzione BSR è quindi affiancata dalla RTS (ReTurn from Subroutine) che si comporta esattamente come l'intuito suggerisce. Questa istruzione non richiede parametri di sorta. BRA e BSR utilizzano sempre offset relativi al program counter, cioè saltano tot byte in avanti o all'indietro rispetto alla posizione attuale, ma cosa accade se per qualche motivo vogliamo saltare a una locazione fissa o anche variabile ma comunque non necessariamente correlata a quella attuale? Le istruzioni JMP (JuMP) e JSR (Jump to SubRoutine) accorrono in nostro aiuto. Queste direttive ammettono come parametro un indirizzo effettivo invece di un offset; sono quindi possibili linee come JMP 10000, che salta alla locazione 10000, o JSR (A2), che richiama una subroutine all'indirizzo contenuto in A2. Sono evidentemente possibili salti calcolati e vere e proprie tabelle di salto, simili all'ON GOTO del Basic. Il JMP è tanto flessibile da poter sostituire il BRA, semplicemente utilizzando come indirizzo effettivo un modo di indirizzamento relativo al program counter: BRA label è equivalente a JMP label(PC). Tuttavia il JMP non possiede una forma corta. Programmando l'Amiga capita spesso di richiamare le routine del sistema operativo con la forma JSR offset(A6).

Il concetto di flag.

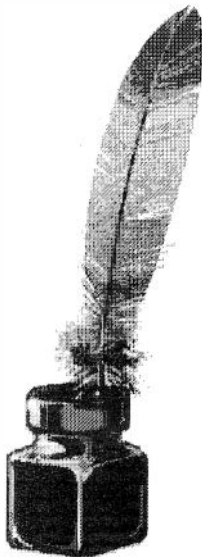
I salti condizionati implicano il concetto di condizione. Cos'è una condizione? Questa domanda apparentemente banale trova la propria ragione d'essere nella duplice constatazione che in quasi tutti i linguaggi evoluti le condizioni vengono espresse tramite espressioni booleane (quelle con AND, OR eccetera) e che in Assembly le espressioni praticamente non esistono. Ebbene anche in questo caso, come già si era visto nel caso degli operandi di una istruzione, l'Assembly prevede una ristrettissima gamma di forme che una condizione può assumere, illustrata in tabella 1. La prima colonna riporta la sigla con la quale l'assembler identifica ogni condizione; si tratta di una coppia di lettere che devono essere incorporate nel nome dell'istruzione stessa. Supponiamo di voler eseguire un salto solo se un certo numero è uguale a zero: l'istruzione adatta è BEQ, dove le ultime due lettere costituiscono la condizione. La forma generica di un salto condizionato è Bcc, dove cc sta per condition code ovvero codice di condizione. Potremo quindi avere istruzioni come BNE, BHI, BLE, BCC e persino BT (Branch if True o Branch Always) che è

Codice	Pronuncia estesa	Espressione booleana
T	True (always)	1
F	False (never)	0
HI	High	not C and not Z
LS	Low or same	C or Z
CC	Carry clear	not C
CS	Carry set	C
NE	Not equal	not Z
EQ	Equal	Z
VC	Overflow clear	not V
VS	Owerflow set	V
PL	Plus	not N
MI	Minus	N
GE	Greater or equal	N and V or not N and not V
LT	Less than	N and not V or not N and V
GT	Greater than	(N and V or not N and not V) and not Z
LE	Less or equal	N and not V or not N and V or Z

Tabella 1: elenco delle possibili condizioni.

Tabella 1: elenco delle possibili condizioni

sinonimo di BRA, mentre BF (Branch if False o Branch Never) non esiste per motivi tecnici (il codice operativo di questa ipotetica istruzione è stato preso a prestito per identificare BSR) ma, tutto sommato, non se ne sente troppo la mancanza essendo un'istruzione che evidentemente, se anche esistesse, non si sentirebbe molto in vena di saltare.

La linea BEQ.S Salve indica che quando un certo numero sarà uguale a zero dovrà avvenire un salto alla label Salve, ma da nessuna parte della linea traspare la più piccola indicazione circa l'identità di chi dovrà essere confrontato con lo zero. Dove si è nascosto? Il fatto è che il confronto è retroattivo. Il salto avverrà se e solo se il risultato dell'ultima operazione effettuata sarà stato uguale a zero. Per esempio la seguente coppia di linee:

```
ADD.W D1,D2
BEQ.S SALVE
```

provoca un salto se la somma di D1 con D2 ha avuto risultato nullo. Tutto ciò potrebbe sembrare strano. Come fa il BEQ a conoscere l'esito dell'istruzione precedente? Questa informazione viene memorizzata in cinque bit del microprocessore chiamati flag di condizione o semplicemente flag; trattasi dei cinque bit meno significativi dello status register, la cui esistenza è stata accennata nella prima lezione. Tramite questi bit il processore ricorda alcune caratteristiche del risultato dell'istruzione precedente; vediamoli più in dettaglio.

Il primo flag si chiama extension e viene indicato con una X; esso è l'unico che non viene mai utilizzato direttamente nelle condizioni, che invece adoperano il flag C che assume quasi sempre lo stesso valore di X. Il flag di estensione viene usato per memorizzarvi il riporto proveniente da istruzioni come ADD o SUB e consente quindi di portare a termine operazioni in precisione multipla suddividendole in un certo numero di operazioni più piccole. Per esempio si può realizzare una somma a 64 bit per mezzo di due addizioni a 32: nella seconda dovrà essere utilizzato il riporto proveniente dalla prima. Esiste un'istruzione (ADDX) che oltre a sommare gli operandi forniti vi aggiunge anche il flag di estensione, realizzando così il riporto. Non si tratta comunque di una tecnica diffusissima, essendo la lunghezza dei registri a 32 bit del 68000 già di tutto rispetto; tale metodo è invece d'obbligo lavorando con processori a otto o sedici bit 'puri'. La tecnica del riporto è invece impiegata di frequente per confrontare tra loro numeri senza segno; se sottraendo il primo dal secondo si ha un riporto significa che il primo era maggiore del secondo.

Il secondo flag si chiama negative e si indica con la lettera N. Altri microprocessori lo possiedono ma sovente lo indicano diversamente; a esempio lo Z80 lo chiama S (segno). Questo flag viene settato (cioè posto a uno) se il risultato dell'operazione (inteso come numero dotato di segno) è negativo. In caso contrario (risultato nullo o positivo) il flag è resettato. Per questo motivo spesso nel gergo dell'Assembly viene chiamata positiva una grandezza che è maggiore o anche soltanto uguale a zero. In parole povere, per il microprocessore lo zero è da considerarsi positivo. Questa convenzione, sostenuta più che altro da un tacito accordo, non provoca solitamente equivoci.

Il terzo flag si chiama zero e viene indicato con una Z. È senz'altro il flag di uso più comune e viene settato solo da un risultato nullo.

Il quarto flag si chiama overflow e si indica con una V. Viene settato quando nel

Confronto tra due numeri CMP A,B			Test di un numero TST A		
Tipo	Con segno	Senza segno	Tipo	Con segno	Senza segno
B=a	EQ	EQ	A=0	EQ	EQ
B<>A	NE	NE	A<>0	NE	NE
B<A	LT	CS (LO)	A<0	MI (LT)	
B<=A	LE	LS	A<=0	(LE)	
B>A	GT	HI	A>0	(GT)	
B>=A	GE	CC (LS)	A>=0	PL (GE)	

Tabella 2: uso appropriato delle condizioni.

Tabella 2: uso appropriato delle condizioni

corso di un'operazione tra numeri dotati di segno il risultato eccede le dimensioni del supporto fisico destinato a memorizzarlo. Per esempio una somma come 100+100 eseguita su byte provoca un overflow, dal momento che 200 non è compreso nel range da -128 a +127. La stessa addizione, è bene notarlo, non setta il flag di estensione, perché 200 non esce dal range che va da 0 a 255.

Il quinto flag si chiama carry (riporto) e si indica con una C. Sotto certi aspetti si comporta allo stesso modo del flag di estensione: tutte le volte che il flag X viene alterato anche C viene aggiornato di conseguenza. Il fatto è che poi il carry tende a perdere il proprio stato più facilmente dell'estensione: basta per esempio un banalissimo MOVE per azzerare il carry, mentre il flag X resta immutato. I motivi che hanno spinto i progettisti del 68000 a sdoppiare il flag di carry sarebbero alquanto complessi da spiegare; l'intenzione comunque era di agevolare il programmatore. Il flag di carry (o estensione che dir si voglia: nel seguito, a meno di casi particolari, questi due termini saranno usati indifferentemente, data la quasi totale identità di comportamento dei due flag) può essere usato per accorgersi se il risultato di un'operazione tra numeri privi di segno trabocca dall'operando destinazione, così come il flag di overflow veniva usato in simili circostanze con i numeri dotati di segno. Per esempio 200+200, su dimensione byte, provoca un riporto dal momento che 400 supera 255. Nella destinazione verrà invece scritto il numero 144 (ossia 400 modulo 256). È bene notare che la stessa operazione NON attiva affatto il flag di overflow, come risulterà chiaro in seguito.

Il complemento a due

Molto spesso in precedenza si è parlato di numeri con segno e senza segno. Che differenza c'è tra un numero senza segno e uno positivo? Come distingue il processore questi due tipi di dati? Ebbene, nella maggior parte dei casi non li distingue affatto. Facciamo un po' d'ordine.

Un numero viene visto dalla CPU (Centrai Processing Unit, sinonimo di processore) come una stringa di cifre binarie; nel caso di numeri a otto bit, che saranno sempre usati nei prossimi esempi, le cifre binarie sono, appunto, otto. Con questi otto bit si possono rappresentare numeri da zero (%00000000) a 255 (%11111111). Tutte le volte che in una operazione su byte si ottiene un risultato con più di otto bit il processore tronca tutti i bit dal nono in poi. Ma come si fa a rappresentare un numero negativo? Ebbene, qual'è la definizione di meno uno? Trattasi di quel numero che, sommato a uno, dà zero. È immediato verificare che 255 risponde alla definizione. Sommando 255 e 1 si ottiene 256=%100000000 che viene troncato e ridotto a zero. Un trucco sleale, penserà qualcuno, e che funziona solo a causa di un difetto comune a tutti i processori, cioè quello di non avere registri di lunghezza infinita e di troncare tutto ciò che ne esce. Forse è così, ma di fatto funziona. Anzi, funziona così bene che si è pensato di ufficializzare questa rappresentazione con una convenzione: si considerano negativi quei numeri il cui bit più significativo è posto a uno. In questo modo i numeri da 128 a 255 ricevono il nuovo significato di numeri tra -128 e -1, rispettivamente. Analogo discorso vale con parole di lunghezza maggiore; usando 16 bit si chiamano negativi i numeri che normalmente sarebbero compresi tra 32768 e 65535 e che vengono adesso, loro malgrado, a trovarsi inclusi a viva forza nel range tra -32768 e -1. Questa rappresentazione si chiama 'complemento a duè e si affianca a quella usata in precedenza per i numeri privi di segno. Naturalmente, a causa del modo stesso in cui si è pensato di definire i numeri negativi, le operazioni di addizione e sottrazione lavorano indifferentemente con numeri dotati o privi di segno, senza che neppure il processore debba esserne informato. Moltiplicazioni e divisioni, al contrario, devono essere definite diversamente a seconda dei casi e richiedono istruzioni separate. Per esempio la moltiplicazione del 68000 esiste in due varianti, MULS e MULU dove la lettera finale significa rispettivamente signed e unsigned. Queste operazioni moltiplicano tra loro numeri a 16 bit fornendo un risultato a 32.

```
SLEEP      MOVEQ #0,D0              ;Inizializza D0.
           MOVEQ #0,D1              ;Inizializza D1.
LOOP       MOVE.B (A0)+,D0          ;Leggi un carattere:
           CMP.B #10,D0             ;e' un linefeed (CHR$(10))?
           BEQ.S EXITLOOP           ;Se si', esci dal loop, altrimenti
           SUB.B #'0',D0            ;sottrai a D0 il codice ASCII di 0,
           MULU #10,D1              ;moltiplica D1 per 10,
           ADD.L D0,D1              ;somma D0 e D1
           BRA.S LOOP               ;e salta a LOOP.
EXITLOOP   LEA $DFF000,A2           ;Carica in A2 l'indirizzo $DFF000,
           MOVE.W #$4000,$9A(A2)    ;disabilita gli interrupt,
           MOVE.W #$0200,$96(A2)    ;interrompi ogni DMA,
           MOVE.W #$000,$180(A2)    ;schermo nero,
           MOVE.B #$FD,$BFD100      ;ferma il motore
           MOVE.B #$F5,$BFD100      ;del drive df0: e
           MOVE.B #$02,$BFE001      ;spegni il led del 'power'.
           MOVEQ #0,D0              ;Inizializza D0.
WAIT       DBRA D0,WAIT             ;Primo loop a vuoto.
           DBRA D1,WAIT             ;Secondo loop a vuoto.
           MOVE.B #$00,$BFE001      ;Riaccendi il led,
           MOVE.W #$8200,$96(A2)    ;riattiva il DMA,
           MOVE.W #$C000,$9A(A2)    ;riabilita gli interrupt,
           MOVEQ #0,D0              ;segnala 'nessun errore'
           RTS                      ;ed esci dal programma.
```

Quando si esegue per esempio una somma il processore altera tutti i flag senza curarsi del fatto che la cosa abbia senso o meno; anche addizionando numeri privi di segno il processore altera il flag di negativo (che è sempre uguale al bit più significativo del risultato) e quello di overflow, che hanno senso solo con numeri in complemento a due, e viceversa. Si può infine capire perché sommando 200 con 200 non si verifica alcun overflow: in complemento a due 200 significa -56 e il risultato della somma è -112, che non esce dal range -128 - +127.

I salti condizionati

L'istruzione CMP (CoMPare) confronta tra loro due operandi (per la precisione sottrae il primo dal secondo, ma senza alterare niente: il risultato viene usato per alterare i flag ma viene poi gettato via). Se i due valori sono uguali la sottrazione dà zero e il flag corrispondente viene settato. Ciò spiega perché la condizione di risultato nullo venga chiamata EQ; ciò deriva dal fatto che molti salti vengono effettuati subito dopo un CMP e la sigla EQ serve a ricordare che un risultato nullo implica l'uguaglianza degli operandi del CMP. Molte sigle di codici di condizione traggono origine da questo fatto. Sintassi possibili: CMP ea,Dn - CMP ea,An - CMP #n,ea. L'istruzione TST esegue un test di un singolo operando (potremmo dire che lo confronta con lo zero); questa istruzione cancella sempre il flag di carry e quello di overflow. Sintassi: TST ea.

Esistono altre tre istruzioni oltre a Bcc

che sfruttano i flag: DBcc, la poco comune Scc e la rarissima TRAPV. La più importante delle tre è senza alcun dubbio la prima, usatissima nella realizzazione dei loop. Iniziamo per semplicità dalla sua forma quasi incondizionata DBRA (Decrement and Branch); la sintassi completa è DBRA Dn,label e occupa quattro byte. Funzionamento: il registro Dn viene decrementato di un'unità e se il risultato è diverso da meno uno il processore salta alla label specificata. Si tratta dunque di qualcosa di concettualmente simile al ciclo FOR-NEXT del Basic, con la differenza che lo step è sempre pari a meno uno e il valore limite è sempre lo zero. Chiariamo le cose con un esempio:

```
        MOVE.W 30000,D0
        MOVE.L #50000,A0
        MOVEQ #0,D1
        BRA.S END
LOOP    ADD.W (A0)+,D1
END     DBRA D0,LOOP
```

Lo scopo del programmino di esempio consiste nel porre in D1 la somma delle N word consecutive che si trovano a partire dalla locazione 50000 in poi, dove N è un numero che si trova memorizzato all'indirizzo 30000. Alcune osservazioni sembrano opportune. Innanzitutto occorre sottolineare ancora una volta la differenza tra un indirizzo e il dato che si trova memorizzato a quell'indirizzo, espressa dalla presenza del carattere '#'. Mentre la prima linea pone in D0 il contenuto della locazione 30000, la seconda scrive in A0 l'indirizzo 50000 inteso come semplice numero. Si puo' poi constatare come l'istruzione MOVEQ sia molto usata nella pratica per predisporre velocemente un piccolo numero in un registro dati ed è sempre sfruttata, in genere, per azzerarlo interamente. Il postincremento del registro A0 consente di prendere in considerazione una dopo l'altra tutte le word di interesse; il caso in cui i dati con i quali bisogna lavorare sono disposti sequenzialmente in memoria è più diffuso di quanto non si tenda a credere e spesso il postincremento risulta davvero comodissimo.

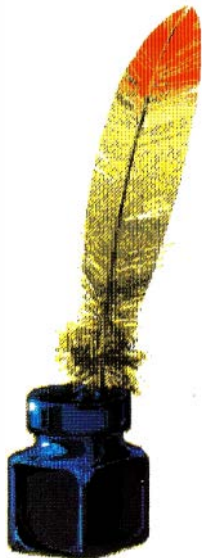
Non si può fare a meno di notare la bizzarra presenza di un branch che salta all'ultima istruzione del loop; a quale scopo? Il problema è che il numero N, se è stato ottenuto in seguito a dei calcoli, potrebbe anche essere nullo. Non dimentichiamo che il ciclo FOR-NEXT esegue subito un test per controllare che il valore iniziale della variabile indice non abbia già superata il valore limite, e se ciò accade esce subito dal loop senza eseguirlo neppure una volta. L'istruzione DBRA è appunto ottimizzata per la realizzazione di cicli nei quali la condizione di uscita viene controllata subito; per sfruttare questa caratteristica basta saltare alla fine del loop. Se il registro dati vale zero l'istruzione, decrementandolo, otterrà meno uno e uscirà subito dal ciclo. Ciò compensa anche il fatto che il DBRA, saltando sul meno uno e non sullo zero, ha la tendenza innata a iterare una volta di troppo: saltando alla fine del loop si elimina la prima iterazione e le cose tornano a posto. Quando invece il numero di iterazioni è fisso, e quindi noto in fase di scrittura programma, la seccatura di aggiungere un BRA può essere evitata, ricordandosi però di inserire nel registro dati un valore inferiore di un'unità al numero di iterazioni previste. Il DBRA decrementa il registro dati su dimensione word.

La generalizzazione del DBRA è DBcc (Decrement and Branch unless condition) che controlla innanzitutto se la condizione specificata si è verificata, nel qual caso esce subito dal loop; se invece la condizione risulta falsa DBcc decrementa il registro e se il risultato è diverso da meno uno salta alla label indicata. DBRA è quindi sinonimo di DBF, come BRA lo era di BT. L'istruzione DBT esiste e non fa assolutamente niente, dal momento che esce subito dal loop in ogni caso. Il DBcc è in sostanza un'istruzione doppiamente condizionata e viene usata, per esempio, quando si deve cercare un dato in una tabella confrontandolo con uno di riferimento; occorre considerare un elemento alla volta e continuare le iterazioni finché il confronto non dà esito positivo oppure si raggiunge la fine della tabella.

Un piccolo programma funzionante

Finalmente esistono le basi necessarie alla presentazione di un programma completo e funzionante, anche se piuttosto breve. La routine si chiama Sleep e compie la pregevole funzione di non fare assolutamente nulla e di costringere l'intero Amiga a fare altrettanto per un certo periodo di tempo. Quando Sleep è attivo l'Amiga sembra spento: tutti i task vengono paralizzati, il drive si ferma all'istante, lo schermo diventa nero e persino la spia dell'alimentazione si spegne. Ottimo per giocare uno scherzo a qualcuno. Peccato solo per la spia del monitor, che non può essere raggiunta via software e resta accesa.

Il programma è semplice ma non banale. Si tratta di un nuovo comando CLI che viene lanciato digitandone il nome in una finestra CLI e che accetta come parametro un numero che rappresenta la durata della stasi misurata approssimativamente in decimi di secondo. Non appena lanciato in esecuzione Sleep deve quindi leggere il numero che l'utente ha digitato dopo il nome (Sleep 100, per esempio) e che è presente in memoria come stringa di codici ASCII puntata dal registro A0 e seguita da un linefeed (=CHR$(10)). Il programma dovrà leggere tale stringa un byte alla volta, tradurre ogni codice ASCII in cifra decimale sottraendogli 48 (le cifre da 0 a 9 hanno codici ASCII compresi tra 48 e 57) e riunire le cifre per ottenere il numero cercato. Il loop di attesa viene poi realizzato con due cicli DBRA concatenati.

Se fate uso del macroassembler standard dell'Amiga prodotto dalla Metacomco dovete seguire la seguente procedura: entrare in CLI, inserire il testo con l'editor ED (esempio: ED Sleep.source), assemblarlo (esempio: ASSEM-DEVEL:c/ASSEM Sleep.source -O Sleep.code) e linkarlo (esempio: ASSEM-DEVEL:c/ALINK Sleep.code TO sys:c/Sleep). Il risultato, se tutto è andato bene, sarà un comando eseguibile da CLI.

GIOCARE CON IL CONTROLLER DEL DISCO

di Fabio Biancotto

Questo articolo tratterà di come l'hardware legge e scrive sul disco e di come il programmatore possa scrivere informazioni irregolari sul disco: incluso come scrive 1120 Kbyte di data su un disco standard dell'Amiga.

Il controller del disco dell'Amiga è interessante perché provvede solo al minimo controllo dell'hardware del disk drive. L'hardware del controller del disco legge e scrive grossolanamente i dati direttamente dal disco, lasciando al programmatore l'interpretazione del significato. La Commodore, naturalmente ha dedicato alcune routine per manipolare il disk, ma qualsiasi hacker vorrà sapere come usarle da solo

Per scrivere sul dischetto, al disk controller viene fornito un indirizzo nella memoria e un word count. Le parole puntate da questo indirizzo sono inviate al disk drive in modo seriale con il primo bit significativo. Il disk drive risponde ai data non facendo nulla per i bit a "0" e invertendo il campo magnetico nelle testine di scrittura per i bit a "1". Il controller può essere settato per mandare i dati a una velocità di 1 bit ogni 2 microsecondi o ogni 4 microsecondi. Considereremo solo il caso ogni 2 microsecondi.

Il disco lavora a una velocità di 300 giri per minuto, o 5 giri per secondo, che significa 200 millisecondi per giro. Perciò con 2 microsecondi per bit, avremo 100 mila bit per giro e scriveremo: 100 mila bit per traccia * 160 traccie per disk = 16 milioni di bit per disk = 2 Mbyte per disk. Sfortunatamente non si può scrivere sul disk tutti questi dati. Ci sono due limitazioni: la prima, non ci possono essere due flussi inversi nella stessa fila (non si può scrivere due "1" nella stessa fila); la seconda, non ci possono essere più di quattro "0" nella stessa fila.

La prima restrizione è causata dal fatto che il campo magnetico non può essere cambiato abbastanza rapidamente per riconoscere due inversioni chiuse contemporaneamente. La seconda limitazione è perché il drive usa il flusso transitorio per misurare la velocità alla quale il drive ruota e aggiusta il clock per compensare le variazioni. Il metodo, ormai standard, per sopperire a queste limitazioni è chiamato MFM encoding. L'MFM encoding converte ogni bit reale in due bit di dati immagazzinati. Un 1 diventa 01 e uno 0 diventa 10 oppure 00, questo dipende dal bit anteriore: se il bit precedente è un 1 allora diventa 00, se è invece uno 0 allora si trasforma in 10. Mentre:

```
AB -> XACB
00->?010
01 -> ?001
10->0100
11 ->0101
```

Logicamente, tra i due bit A e B inseriamo un bit C tale che:

C =~ (A/B) = NOT (A OR B)

Notare che i secondo bit è sempre lo stesso data bit. La prima restrizione è giustificata perché 01 e seguito da 01 o 00, e 11 non è permesso, e 10 deve essere preceduto da 00 o 10. La seconda restrizione è giustificata perché di zeri principali non ne possiamo avere più di tre nella stessa fila (nel modello 101 - 010001).

Nell'Amiga, la conversione dai data reali a quelli immagazzinati, viene fatta dal software. In alcuni computer, per esempio l'IBM PC, questa conversione è fatta dall'hardware. Per fortuna, l'Amiga ha il blitter. Usando il blitter, i data possono essere convertiti dal formato MFM molto velocemente. Dopo aver immagazzinato due bit di data per ogni bit reale, il MFM può essere formato molto facilmente se ha riversato i dati reali in due serie, i bit pari e

i bit dispari. I bit dispari sono sempre altri bit partiti con il primo, i pari sono tutti gli altri. I bit pari e dispari sono immagazzinati separatamente sul disk. Questo semplifica la conversione perché i bit lasciano spazi vuoti nel data, nei quali possono essere inseriti altri extra bit ordinati dal MFM. Per esempio i data:

01100101 - pari x1x0x1x1 e dispari x0x1x0x0 - pari x1000101 e dispari x0010010

L'encoding per ogni block può essere fatto in tre passate del blitter. Il primo passo è per cercare il nuovo bit che sarà inserito tra i bit reali. Questo è fatto per ambedue i bit pari e dispari alla volta. Il minterm è un blitter di carattere tecnico:

```
Blitter input  A – i dati reali sono shiftati a
                   destra di 2 bit
               B – dati reali
               C – niente
Blitter output D -- ~(A/B)minterm = $11
```

Il secondo passo fonde i dati reali pari con il risultato del primo passo:

```
Blitter input  A -- passo primo D shiftato a
                    sinistra di 1 bit
               B -- dati reali
               C – $5555 = 0101010101010101
Blitter output D – (B & C)/(A & ~C)minterm =
                   $CA
```

Passo terzo fonde i dati reali dispari con il risultato del primo passo:

```
Blitter input  A -- i dati reali sono shiftati a
                    destra di 1 bit
               B -- passo primo D
               C– $5555 = 0101010101010101
Blitter output D -- (A & C)/(B & ~C)minterm =
                    $AC
```

Alcuni bit alla fine devono essere ancora aggiustati. Il tempo usato dal blitter per fare l'encoding può essere stimato dal numero di riferimenti dinecessari: 11 cicli di memoria per parola o ancora 20 millisecondi per traccia. Da un minimo di 200 millisecondi, più il tempo per cercare la testata, per scrivere nel disk, ciò non interessa la velocità di lettura del disk. Quando il disco legge i data deve decodificarli. La decodificazione è semplificata dall'encoding e può essere fatta in una sola operazione di blitter.

```
Blitter input  A – i dati immagazzinati sono
                   shiftati a sinistra di uno
               B -- dati immagazzinati
               C -- $5555 = 0101010101010101
Blitter output D – (A & ~C)/(B & C)minterm =
                   $CA
```

Questo può usare all'incirca quattro riferimenti di memoria per parola, che è all'incirca 8 millisecondi per traccia.

Il formato di ricodificazione dei dati per settori dell'AmigaDOS è descritto nel manuale ROM Kernal volume 2. Non possiamo sapere se la Commodore attualmente usa il blitter nel modo che vi abbiamo descritto, ma non ne abbiamo visti di più veloci. Usando l'MFM immagazziniamo 50 mila bit per traccia che equivalgono a 1 Mbyte per disk. Ahimè, non possiamo però prelevare una tale quantità di dati dal disco, perché questo non deve girare troppo velocemente. Se girasse, infatti molto velocemente e tenteremmo di scrivere tutti i 50 mila bit, gli ultimi bit verrebbero scritti sopra i primi bit e perderemmo qualche data. Un incremento del 1% di velocità ci farebbe sovrapporre 500 bit. La Commodore attualmente immaganizza 47888 bit ogni traccia, dei quali sono usati per data 45056. Mentre l'MFM encoding soddisfa le restrizioni elencate, ci sono altri schemi di encoding che immagazzinano molti data per traccia e soddisfano maggiormente le limitazioni elencate precedentemente.

Per prima, noteremo che la lunghezza arbitrariamente permessa al bit campione può essere costituita ordinando le stringhe dei bit 01, 001, 0001 e 00001 in sequenze arbitrarie. Possiamo calcolare il numero di campioni di una data lunghezza che possonon essere fatti ricorrendo a questo modo. Per 2 e 3 c'è un campione, per 4 ci sono due, 0001 e 0101; per 5 ci sono tre, 00001, 01001 e 00101. In genere si può calcolare il numero di campioni Xn, per n bit dalla formula

$$X_n = X_{n-2} + X_{n-3} + X_{n-4} + X_{n-5}$$

Il termine Xn-2 è seguito da 01 per tutte le configurazioni di lunghezza n-2, il termine Xn-3 è seguito da 001 per tutte le configurazioni di lunghezza n-3, eccetera.

nXn	
21	
3	1
5	3
64	
77	
8	10
9	16
10	24
11	37
12	57
13	87
14	134
15	205

Questi modelli sono troppo restrittivi perché incominciano tutti con 0 e finiscono con 1. Possiamo accettare modelli che ci permettano di avere degi zeri in coda. Per un motivo che sarà chiarito più avanti possiamo permetterci di avere soltanto 3 zeri in coda al nostro campione. Definiamo ora Pn:

$$P_n = X_n + X_{n-1} + X_{n-2} + X_{n-3} = X_{n+2}$$

Che è il numero di campioni senza zeri più il numero con un solo zero più il numero con 2 e 3 zeri in coda. Abbiamo ora un pezzetto di stringa contenente da 1 a 4 zeri in testa e da 0 a 3 zeri in coda. Questa stringa può essere messa unitamente dentro una stringa accettabile; se si adotta la regola che il primo bit in ogni campione diventi un 1 se vicino ci sono degli zeri. Questo è lo stesso del MFM per n=2, P2=X4=2. I 2 campioni sono 01 e 00. Ci sono altri due n di particolare interesse. Per n=7, P7=X9=16; così si possono ricodificare 4 bit di data reali dentro 7 bit di data immagazzinati (mentre è comparato con 8 bit di data immagazzinati con l'MFM). I conmpioni sono:

?001000?000100?010100?010010
?001010?010001?010101?001001
?000101010100001001000100010
0101010010000101010010100101

Il "?" sarà 1 se preceduto da 7 bit che termineranno con uno 0. Questo può essere assegnato arbitrariamente al sedicesimo 4 bit campione. Il secondo caso interessante è n=12, P12=X14=134; si possono codificare 7 bit di data reali dentro 12 bit di data immagazzinati (opposti ai 14 dell'MFM). Compiere l'encoding per questo metodo è molto più difficile della stessa operazione fatta per l'MFM encoding. Il 7/12 potrebbe essere ricodificato usando il blitter per separare i data reali in pezzi di 7 bit, quindi si può usare la tabella per cercare il corrispondente codice di 12 bit e, finalmente, usando ancora il blitter, fondere il codice di 12 bit in un blocco continuo. Questo processo sarà fatto in circa 70-80 millisecondi per traccia.

Il blitter deve essere usato per versare i dati immagazzinati dentro le parti di 12 bit, allora la tabella è usata per cercare il corrispondente di 7 bit, e i pezzi di 7 bit devono essere fusi ancora assieme per recuperare il data reale. Ricodificare porterà via ancora 70-80 millisecondi per traccia. Usando il codice 7/12 possiamo nominalmente immagazzinare 7/12 * 100 mila = 58333 bit per traccia. Sottraendone alcuni per prevenire sovrapposizioni possiamo immagazzinare 56192 bit per traccia che sono 7 Kbyte per traccia o 1120 Kbyte per disk. Se il disk è diviso in settori nel modo Commodore, potremmo riempire 13 settore per traccia e avere 1040 Kbyte per disk di data usabili. Questo rappresenta un incremento del 18% superiore ai 880 Kbyte per disk del MFM encoding.

CORSO DI AMIGA BASIC

Terza puntata: il menù e il controllo degli utenti

di Paolo Russo

All'inizio non esistevano. Sono apparsi gradualmente, senza rumore, quasi senza che nessuno se ne accorgesse; erano rozzi e dovevano essere gestiti dal programmatore con molta fatica e poca soddisfazione. A poco a poco si sono evoluti, diventando meglio gestibili, più facilmente utilizzabili, esteticamente accattivanti e, perché no, perfino più discreti. Nell'arco di pochi anni sono stati sviluppati sistemi operativi in grado di supportarli degnamente e sono finiti col diventare una componente ineliminabile della nostra realtà informatica. Sono attualmente assai popolari presso tutti gli utenti di package applicativi che non hanno voglia di leggersi i manuali. Sono i menu, uno dei punti di forza della filosofia WIMP (Windows, Icons, Mouse, Pull-down menus) che l'Amiga ha fatto propria. L'AmigaBASIC li supporta in maniera non del tutto completa ma comunque più che sufficiente per le normali applicazioni e soprattutto in modo molto programmer-friendly, intendendo con questa affermazione che il programmatore medio può crearli e gestirli facilmente, al contrario di quanto avviene con altri linguaggi come il C che li supportano in modo molto esteso ma anche intricato e decisamente scomodo. L'argomento menu è poi collegato alla più generale problematica del controllo degli eventi.

Natura del menu

Dal punto di vista dell'utente un menu è una cosa che scende dall'alto quando si preme il tasto di destra del mouse e che consente di lanciare comandi, scegliere opzioni e predisporre parametri di vario genere. Dal punto di vista del programmatore i menu sono entità gestibili tramite una serie di comandi, quasi tutti basati sulla parole chiave MENU; la visualizzazione dei menu ogni volta che si preme il menu button è gestita automaticamente da Intuition senza bisogna di interventi da parte del programma, come pure l'evidenziazione delle singole opzioni sulle quali passa il pointer e la scomparsa del menu con ripristino dello sfondo preesistente quando il tasto viene lasciato. Il programma viene impegnato solo all'inizio, quando deve definire la struttura dei menu, e viene poi scomodato nei momenti più impensati da Intuition ogni volta che l'utente ha selezionato qualcosa. Accade infatti che Intuition lanci dei segnali di varia natura per avvertire dell'avvenuta selezione ed esistono almeno un paio di modi per accorgersi di questi messaggi.

Creazione dei menu

Il comando per creare un menu è MENU numero,opzione,stato(,stringa) che deve essere usato ripetutamente per definire ogni singola opzione di ogni singolo menu. Il parametro numero caratterizza il menu sul quale il comando avrà effetto; i menu sono infatti numerati da uno a dieci. Quello dotato di numero inferiore compare all'estremo sinistro dello schermo e tutti gli altri seguono in ordine numerico. Naturalmente non è obbligatorio avere dieci menu; se ne possono avere molti di meno, l'interprete BASIC per esempio ne usa quattro. Opzione rappresenta il numero, da 0 a 19, della singola opzione all'interno del menu specificato, con la particolarità che solo i valori da uno in poi sono associati a effettive opzioni mentre il valore zero contrassegna il menu tutto intero, o meglio il suo titolo che appare nella barra superiore premendo il tasto dei menu. Per creare un menu occorre innanzitutto crearne il titolo; per esempio il comando MENU 1,0,1,"Settimana" predispone un menu (per il momento privo di opzioni) all'estremo sinistro della barra superiore. Premendo il menu button la scritta Settimana diventerà visibile. È interessante notare che il suddetto menu si sostituisce al primo menu dell'AmigaBASIC, quello di nome Project. Adesso bisogna aggiungere le opzioni, con una serie di linee del tipo:

```
MENU 1,1,1,"Lunedì"
MENU 1,2,1,"Martedì"
MENU 1,3,1,"Mercoledì"
```

Da adesso in avanti il menu verrà mostrato automaticamente da Intuition ogni volta che l'utente cercherà di accedervi. Possiamo creare altri menu, da affiancare al precedente, impiegando la stessa tec-

nica ma usando un numero diverso da uno come primo parametro. Ma cosa accade poi quando l'utente non si accontenta di giocherellare con il mouse ed effettua una selezione?

Controllo dei menu

Esiste una funzione, MENU(0), che normalmente restituisce sempre zero ma occasionalmente, quando cioè si è verificata una selezione, fornisce il numero del menu contenente l'opzione scelta. È senz'altro opportuno immagazzinare il valore fornito da questa funzione in una variabile, in quanto questo valore compare una sola volta. Se dopo una selezione dal primo menu eseguite per due volte di seguito un PRINT MENU(0) otterrete prima uno e poi zero. Comunque, questa strana funzione non è così smemorata come sembra; se interrogata anche mezz'ora dopo l'atto della selezione, MENU(0) riporterà regolarmente (anche se per una sola volta) il numero del menu selezionato. Il motivo della laconicità di questa funzione, che parla una volta sola e poi tace senza rimedio, affonda le sue radici nel fatto che se il valore permanesse nelle successive chiamate della funzione non sarebbe possibile accorgersi delle ulteriori selezioni effettuate dall'utente magari all'interno dello stesso menu. In questo modo, invece, la funzione assume un valore non nullo solo ogni volta che viene effettuata una selezione. La cosa è, in pratica, molto semplice e comoda da usare.

MENU(1) fornisce invece il numero dell'opzione scelta nell'ambito del menu e non soffre dei problemi di amnesia tipici di MENU(0). Si può quindi pensare di realizzare un loop che legge in continuazione la funzione MENU(0) per accorgersi delle selezioni (polling) e dal quale il programma esce soltanto per portare a termine l'operazione richiesta dall'utente. Ciò è un tantino riprovevole, perché significa sprecare il tempo del processore che si trova costretto a eseguire loop a vuoto e non è disponibile per altri task, e in diversi casi può non essere la soluzione ottimale per altri motivi. Per esempio si può desiderare che un programma di animazione funzioni con continuità lasciando però all'utente la possibilità di intervenire sui menu.

Il controllo degli eventi

La selezione di un'opzione da parte dell'utente è un evento. Altri tipi di eventi possono essere il tentativo di fermare il programma con un CTRL-C, un errore qualunque in fase di esecuzione, la pressione del tasto sinistro del mouse, la collisione di due sprite o lo scadere di un certo intervallo di tempo. Esistono nel sistema Amiga altri tipi di eventi oltre a quelli elencati, come l'inserzione di un disco in un drive o la pressione di un tasto o ancora il movimento stesso del mouse, ma non sono supportati dall'interprete. Gli eventi possono essere controllati con la statement ON evento GOSUB subroutine, dove evento può essere una delle seguenti parole: MENU, BREAK, ERROR, MOUSE, COLLISION, TIMER(n). Nel caso di ERROR, a essere pignoli, bisogna usare ON ERROR GOTO e non GOSUB e la subroutine di gestione deve terminare con RESUME e non RETURN. Questa differenza è dovuta al fatto che un errore non è esattamente un evento che venga determinato da cause esterne al programma stesso o che possa essere ignorato. Il comando ON evento GOSUB predispone una routine per gestire l'evento in questione, ma questo di per sè non ha effetto apparente. Occorre attivare la gestione dell'evento con la sintassi evento ON.

Nel caso del controllo dei menu, dopo aver predisposto una routine di gestione con ON MENU GOSUB subroutine occorre quindi anche un MENU ON. Tutte queste direttive sono reversibili: evento ON viene annullato da evento OFF e ON evento GOSUB subroutine da ON evento GOSUB 0. Dopo aver eseguito MENU ON ogni volta che l'utente effettuerà una selezione la routine indicata nella linea ON MENU GOSUB verrà automaticamente richiamata, qualunque cosa stia facendo in quel momento il programma principale. Il comando MENU STOP ferma temporaneamente la gestione del menu; l'unica differenza tra MENU OFF e MENU STOP consiste nel fatto che quest'ultimo impone all'interprete di ricordare un evento. Se quindi dopo un MENU STOP l'utente seleziona qualcosa e poi il programma esegue un MENU ON si verificherà un salto immediato alla routine di gestione, in quanto l'interprete durante la fase di sospensione dei menu ha ugualmente preso nota della selezione. Comunque in queste condizioni viene ricordata al massimo una singola selezione.

Checkmark e ghosting

Come tutti certamente sanno, nell'Amiga si aggirano spettri. Come, non li avete mai visti? Subito dopo il caricamento dello Workbench provate a selezionare l'opzione Empty trashcan senza aver prima clickato sul cestino: scoprirete che non solo l'opzione non funziona, ma è sbiadita e non si riesce neppure a evidenziarla passandoci sopra. Similmente accade tentando con altre opzioni, come Info, quando nessuna icona è selezionata. Queste opzioni virtualmente inaccessibili sono dette opzioni fantasma (ghost items). In tutti gli esempi precedenti il terzo parametro del comando di creazione dei menu era sempre stato posto uguale a uno; ponendolo a zero l'opzione che viene creata con quel comando diventa fantasma. Si può rendere inaccessibile un intero menu assegnandogli un titolo fantasma (MENU 1,0,0,"Settimana"). Esiste poi il checkmark, quel segno a forma di V asimmetrica che viene spesso collocato vicino al margine sinistro di un'opzione per contrassegnarla come attiva. Per applicare un checkmark a un'opzione il terzo parametro deve assumere il valore due e bisogna ricordare di lasciare almeno due spazi liberi iniziali nella stringa che descrive l'opzione. Non essendo possibile che il terzo parametro assuma contemporaneamente i valori zero e due ne deriva che nessuna opzione può essere fantasma e al tempo stesso avere un checkmark.

Per cambiare lo stato di un'opzione già esistente basta ripetere il comando MENU senza neppure dover riportare la stringa. Per esempio il comando MENU 2,3,0 disabilita la terza opzione del secondo menu. Si può anche cambiare il nome di un'opzione assegnandogli una nuova stringa.

Problemi e trucchi

Supponiamo che, come talvolta capita, si desideri gestire i menu dinamicamente, ossia aggiungendo o togliendo opzioni in fase di run-time. L'aggiunta non presenta problemi, ma per eliminare un'opzione situata magari in mezzo alle altre occorre ridefinire l'intero menu. Infatti ogni volta che si ridefinisce la stringa che fa da titolo a un menu tutte le opzioni sottostanti vengono eliminate come se non fossero mai esistite. Questo trucco serve anche per risolvere un problema che si presenta quando si crea un programma che gestisce meno di quattro menu; in questa situazione i vecchi menu dell'interprete non sono tutti sostituiti dai nuovi e alcuni di essi rimangono, rovinando l'estetica del programma. Per sfrattare questi ospiti indesiderati è sufficiente assegnare loro una stringa nulla per titolo. Un altro problema deriva talvolta dalle differenze di lunghezza tra i vari nomi delle opzioni; se all'interno di uno stesso menu la prima opzione ha un nome di dieci caratteri e la seconda ne ha uno di tre capita che quest'ultima sia più difficile da

selezionare in quanto il mouse deve obbligatoriamente passare su uno dei tre caratteri del nome per evidenziarla. In questo caso può essere utile prolungare il nome con degli spazi vuoti; esteriormente non si nota alcuna differenza ma il mouse riesce adesso a selezionare senza problemi.

Il vostro programma, che fa uso di menu, è entrato in loop irreversibile a causa di qualche problema. Lievemente a disagio vi apprestate a fermarlo con il classico Amiga-punto e scoprite, con assai maggior disagio, che non si ferma affatto. Non lo avevate ancora salvato. Panico. Niente paura, il CTRL-C funziona ancora; Amiga-R, tuttavia, non riesce a farlo ripartire. Cos'è accaduto? Amiga-punto e Amiga-R non sono veri comandi da tastiera, ma semplicemente dei menu shortcut, ossia un modo abbreviato di selezionare opzioni dai menu che non fa uso del mouse ma della tastiera; infatti le opzioni Start e Stop si trovano normalmente in un menu, che il vostro programma ha eliminato sostituendolo con uno suo. Per ripristinare i vecchi menu il comando è MENU RESET.

Un esempio applicativo

Talvolta un listato risulta più eloquente di tante spiegazioni. Il programma Sfere è un esempio di come i menu possano essere vantaggiosamente impiegati perfino nella realizzazione di un programma concettualmente banale come questo, che si limita a disegnare delle pseudo sfere a caso in tutto lo schermo. Il primo menu (Comandi) consente quattro possibilità: dare inizio al disegno, fermarlo prima del termine, chiedere aiuto e uscire dal programma. Il primo di questi comandi lancia in esecuzione la parte principale del programma, che traccia le sfere, mentre tutte le opzioni diventano fantasma, eccezion fatta per Fermati e per il quarto menu che è sempre accessibile anche durante il disegno. Quando non vi è alcun tracciamento in corso l'unica opzione fantasma è Fermati, che risulta priva di senso quando non c'è niente da arrestare. Il comando Aiuto merita una descrizione particolare. Lo scopo di questa direttiva è fornire informazioni all'utente che non sa come usare il programma; attivando Aiuto tutte le opzioni diventano disponibili ma la successiva selezione non lancerà in esecuzione la direttiva corrispondente, bensì visualizzerà sullo schermo alcune righe di spiegazioni riguardanti l'opzione appena indicata. Se quindi l'utente non sa cosa significa, in pratica, Colori casuali dovrà solo selezionare prima Aiuto e poi Colori Casuali per avere una breve spiegazione. Se tutti gli applicativi adottassero metodi simili la vita dell'utente sarebbe notevolmente semplificata.

Il secondo menu consente di cambiare alcuni parametri di tracciamento: il raggio minimo e massimo delle sfere, il loro numero, il numero di linee all'interno di ogni sfera. Il terzo menu contiene un elenco di modi grafici: il programma è infatti appositamente progettato per lavorare con qua-

```
' Sfere by Paolo Russo - Amiga Magazine
CLEAR ,10000
gm=2:rmin=20:rmax=25:nsfere=28:nlin=10
DIM flag(3)

Main:
a=gm-1
IF a AND 4 THEN lx=640:xres=2 ELSE lx=320:xres=1
IF a AND 2 THEN ly=400:yres=2 ELSE ly=200:yres=1
IF a AND 1 THEN ly=CINT(ly*1.28)

RANDOMIZE TIMER
SCREEN 1,lx,ly,3,1-(lx>500)-2*(ly>300)
WINDOW 2,"Sfere",(0,0)-(lx-9,ly-14),0,1
MENU 1,0,1,"Comandi"
MENU 1,1,1,"Disegna"
MENU 1,2,0,"Fermati"
MENU 1,3,1,"AIUTO !"
MENU 1,4,1,"Fine   "
MENU 2,0,1,"Parametri"
MENU 2,1,1,"R min  "
MENU 2,2,1,"R max  "
MENU 2,3,1,"N sfere"
MENU 2,4,1,"N linee"
MENU 3,0,1,"Risoluzione"
MENU 3,1,1,"  320x200"
MENU 3,2,1,"  320x256"
MENU 3,3, 1,"  320x400"
MENU 3,4,1,"  320x512"
MENU 3,5,1,"  640x200"
MENU 3,6,1,"  640x256"
MENU 3,7,1,"  640x400"
MENU 3,8,1,"  640x512"
MENU 3,gm,2
MENU 4,0,1,"Opzioni"
MENU 4,1,1+flag(1),"  Beep alla fine"
MENU 4,2,1+flag(2),"  Loop alla fine"
MENU 4,3,1+flag(3),"  Colori casuali"
ON MENU GOSUB MenuHandler
MENU ON

PALETTE 0,0,0,0
PALETTE 1,1,1,1
FOR i=1 TO 6
  IF i AND 1 THEN b=1 ELSE b=0
  IF i AND 2 THEN r=1 ELSE r=0
  IF i AND 4 THEN g=1 ELSE g=0
  m=(SQR(r*r+g*g+b*b)+1)/2
  PALETTE i+1,r/m,g/m,b/m
NEXT

WHILE endflag=0
  IF flag(2) THEN GOSUB Disegna
  WHILE drawflag=0
    SLEEP
  WEND
  IF endflag=0 THEN
    CLS
    k=xres/yres:pi2=ATN(1)*2
    FOR n=1 TO nsfere
      r=INT(RND*(rmax-rmin)+rmin)
      rx=r*xres
      ry=r*yres
      x=RND*lx
      y=RND*ly
      IF flag(3) THEN c=INT(RND*7) ELSE c=(c+1-(c=0)) MOD 8
      IF c=1 THEN c=0
      FOR i=0 TO ry
        j=SQR(ry*ry-i*i)*k
        LINE (x-j,y-i)-(x+j,y-i),c
        LINE (x-j,y+i)-(x+j,y+i),c
      NEXT
      LINE (x-rx,y)-(x+rx,y)
      LINE (x,y-ry)-(x,y+ry)
      FOR i=1 TO nlin
        s=SIN(i/nlin*pi2)
        a=s/k
        IF a>1 THEN ra=rx*a ELSE ra=rx
        b=1/s/k
        IF b<1 THEN rb=ry/b ELSE rb=ry
        CIRCLE (x,y),ra,,,,a
        CIRCLE (x,y),rb,,,,b
      NEXT
      IF drawflag=0 THEN n=nsfere
    NEXT
    IF flag(1) THEN BEEP
    GOSUB Fermati
  END IF
WEND
WINDOW CLOSE 2
SCREEN CLOSE 1
RUN 1
' SYSTEM

MenuHandler:
m0=MENU(0):m1=MENU(1)
IF help THEN HelpMenu
ON m0 GOTO Comandi,Parametri,Risoluzione,Opzioni

Comandi:
ON m1 GOTO Disegna,Fermati,Aiuto,Fine

Disegna:
drawflag=1
GOTO SetMenu

Fermati:
drawflag=0
SetMenu:
MENU 1,1,1-drawflag
MENU 1,2,drawflag
MENU 1,3,1-drawflag
MENU 1,4,1-drawflag
MENU 2,0,1-drawflag
MENU 3,0,1-drawflag
RETURN

Aiuto:
help=1
MENU 1,2,1
```

```
PRINT"Seleziona l'opzione che non conosci."
RETURN

Fine:
endflag=1:drawflag=1
RETURN

Parametri:
IF m1=1 THEN
  Insert rmin
ELSEIF m1=2 THEN
  Insert rmax
ELSEIF m1=3 THEN
  Insert nsfere
ELSE
  Insert nlin
END IF
RETURN

SUB Insert( var) STATIC
PRINT"Il valore attuale e'";var
LINE INPUT"Inserisci il nuovo valore ";a$
IF a$<>"" THEN var=VAL(a$)
END SUB

Risoluzione:
IF m1=gm THEN RETURN
OPEN"o",1,"ram:sferetemp"
WRITE#1,m1,rmin,rmax,nsfere,nlin,flag(1),flag(2),flag(3)
CLOSE 1
WINDOW CLOSE 2
SCREEN CLOSE 1
RUN 2
2 OPEN"i",1,"ram:sferetemp"
DIM flag(3)
INPUT#1,gm,rmin,rmax,nsfere,nlin,flag(1),flag(2),flag(3)
CLOSE 1
KILL "ram:sferetemp"
GOTO Main

Opzioni:
flag(m1)=1-flag(m1)
MENU 4,m1,1+flag(m1)
RETURN

HelpMenu:
ON m0 GOSUB HelpComandi,HelpParametri,HelpRisoluzione,HelpOpzioni
help=0
MENU 1,2,0
RETURN

HelpComandi:
IF m1=1 THEN
  PRINT"Questo comando da' inizio al disegno"
  PRINT"che e' l'unico scopo di questo"
  PRINT"programma."
ELSEIF m1=2 THEN
  PRINT"Questo comando ferma il disegno, allo"
  PRINT"scopo di cambiare qualche parametro."
ELSEIF m1=3 THEN
  PRINT"Questo comando, come vedi, fornisce"
  PRINT"informazioni sulle possibilita'"
  PRINT"offerte dai menu."
ELSE
  PRINT"Questo comando serve ad uscire dal"
  PRINT"programma e tornare allo Workbench."
END IF
RETURN

HelpParametri:
IF m1=1 OR m1=2 THEN
  PRINT"Il raggio di ogni sfera e' scelto a"
  PRINT"caso tra quello minimo e quello"
  PRINT"massimo. Entrambi i raggi sono"
  PRINT"espressi in pixel di bassa"
  PRINT"risoluzione."
ELSEIF m1=3 THEN
  PRINT"Il numero di sfere che viene"
  PRINT"tracciato sullo schermo ogni volta"
  PRINT"e' sempre lo stesso, se nessuno"
  PRINT"lo cambia."
ELSE
  PRINT"Questo parametro rappresenta la"
  PRINT"meta' del numero di 'meridiani' o"
  PRINT"'paralleli' che vengono tracciati"
  PRINT"per ogni sfera."
END IF
PRINT"Questo parametro puo' essere cambiato"
PRINT"dopo averlo selezionato da menu."
RETURN

HelpRisoluzione:
PRINT"Agendo su questo menu puoi cambiare il"
PRINT"modo grafico impiegato per il disegno."
PRINT"Usando una risoluzione molto alta puo'"
PRINT"accadere che la memoria non sia"
PRINT"sufficiente."
IF m1>6 THEN
  PRINT"Questo modo grafico, in particolare,"
  PRINT"risulta quasi proibitivo."
END IF
RETURN

HelpOpzioni:
IF m1=1 THEN
  PRINT"Questa opzione fa si' che venga"
  PRINT"generato un beep alla fine"
  PRINT"di ogni disegno"
ELSEIF m1=2 THEN
  PRINT"Questa opzione costringe il"
  PRINT"programma a disegnare all'infinito"
  PRINT"senza fermarsi tra un'immagine
  PRINT"e la successiva "
ELSE
  PRINT"Normalmente i colori delle sfere"
  PRINT"sono scelti ciclicamente, ma con"
  PRINT"questa opzione sono casuali."
END IF
RETURN
```

lunque risoluzione (sempre però con otto colori) e l'utente può scegliere il modo grafico che preferisce. L'ultimo menu contiene alcune opzioni che costituiscono un esempio di uso dei checkmark; si può inserire un beep alla fine del tracciamento, far girare il programma all'infinito e cambiare la legge di scelta dei colori.

Alcune parti del programma sono decisamente strane. La routine di tracciamento ellissi utilizza strane relazioni basate su criteri semiempirici per compensare il bizzarro comportamento dell'istruzione CIRCLE che cambia improvvisamente le dimensioni dell'ellisse quando nel corso del ciclo FOR-NEXT il rapporto tra gli assi passa da meno di uno a più di uno o viceversa. Evidentemente la Microsoft ha pensato bene di definire 'raggio' di un ellisse il suo semiasse maggiore, sia esso l'orizzontale o il verticale, con il risultato che quando si disegnano molte ellissi cercando di tenere fisso un semiasse e facendo variare l'altro si ottengono grafici completamente folli.

Un altro problema deriva dalla scarsa propensione dell'interprete alla ridefinizione di una finestra già esistente. L'unico comando capace di eliminare completamente una finestra in vista di una sua riapertura con parametri diversi sembra essere il RUN, che disgraziatamente cancella tutte le variabili. Ecco allora presentarsi un simpatico dilemma: come si fa a cambiare il modo grafico in base alla scelta dell'utente? Occorre chiudere lo schermo e riaprirlo con nuovi parametri, ma così facendo è anche necessario riaprire la finestra contenuta nel suddetto schermo e per farlo occorre dare un RUN che cancella inevitabilmente ogni informazione circa la selezione appena effettuata, così che il programma non ricorda più con quale modo grafico deve riaprire lo schermo. Sfere risolve il problema creando un file temporaneo in RAM disk nel quale memorizza tutte le variabili importanti prima del RUN e dal quale le recupera subito dopo.

Possibili miglioramenti

Sfere vuole essere solo un esempio di gestione di menu. È indubbiamente possibile apportare delle modifiche al programma; per esempio si può migliorare il sistema di autodocumentazione aggiungendo una scorciatoia, cioè consentendo all'utente di premere il tasto Help invece di selezionare Aiuto dal primo menu. Ciò renderebbe più pratico l'uso della funzione di help. Dal punto di vista grafico si può correggere la visualizzazione dei 'meridiani' e dei 'paralleli' che sono correntemente tracciati in uno strano miscuglio di prospettiva approssimata e assonometria; probabilmente un disegno perfettamente corretto avrebbe richiesto un tracciamento per singoli pixel e sarebbe stato micidialmente lento. Non è però esclusa l'esistenza di un metodo geometricamente impeccabile o almeno discretamente approssimato per disegnare queste curve tramite archi di ellisse.

di Paolo Russo

Uno dei quattro punti cardinali della rosa WIMP (Windows, Icons, Mouse, Pulldown menus) è rappresentato dalle icone. Queste simpatiche entità consentono non soltanto di rappresentare graficamente con notevole immediatezza la funzione svolta dai programmi ai quali appartengono ma anche di rendere i programmi stessi facilmente manipolablli; copiare da un supporto a un altro, cambiare nome o directory di appartenenza, cancellare o lanciare in esecuzione un programma dotato di icona sono tutte operazioni a prova di idiota. Ciò che è più importante, comunque, non è tanto la relativa sicurezza offerta da Intuition quanto la praticità e immediatezza di ogni operazione. Clickare o spostare un'icona sono quindi azioni non soltanto riservate agli sciocchi, che si trovano nell'impossibilità di sbagliare, ma adatte a qualunque utente, anche il più smaliziato, che nonostante conosca il CLI come le sue tasche non può negare che sia più veloce e pratico spostare il mouse che digitare una sequenza di caratteri alla tastiera. Dotare di icona un programma non è semplice: occorre capire come funzionano certi meccanismi del sistema operativo, ma una volta compiuto questo sforzo non ne saranno necessari altri.

Struttura di un'icona

L'icona di un programma è memorizzata in un file a parte che porta lo stesso nome del programma associato ma con l'estensione .info in più. Quando ci si trova in ambiente CLI e bisogna trasferire un file dotato di icona ci si deve ricordare di compiere l'operazione prima sul file vero e proprio e poi sulla sua icona. Essenzialmente, il file .info è una struttura di tipo gadget completa delle sue sottostrutture salvata su disco; lo Workbench tratta infatti le sue icone come dei gadget e le gestisce tramite Intuition. È inutile approfondire in questa sede la struttura esatta del file; chi è particolarmente interessato all'argomento e possiede tutti i manuali dell'Amiga può iniziare con l'esame della struttura DiskObject nel file workbench/workbench.h riportata alla pagina 179 dell'appendice D del ROM Kernel Manual: Libraries and Devices, ma si ritroverà ben presto costretto ad aprire simultaneamente tutti i rimanenti manuali nel disperato tentativo di comprendere le sottostrutture. Ciò che importa maggiormente è che ogni icona può appartenere a una sola di diverse classi: tool, project, drawer, trash, disk e altre ancora. Ci occuperemo delle icone di tipo tool che sono caratteristiche dei programmi eseguibili.

Creazione dell'icona

Generare un'icona per il proprio programma è un'operazione molto semplice. L'utility IconEd presente nel disco di sistema consente di creare o modificare un'icona qualunque; se non viene specificato altrimenti, l'icona creata è di tipo tool e questo incontra perfettamente le nostre esigenze. IconEd non ha esattamente la potenza di Deluxe Paint II ma i suoi menu sono discretamente chiari e può essere comunque utilizzato date le ridotte dimensioni del disegno da realizzare. Dopo aver disegnato l'icona occorre salvarla con l'opzione Save e il gadget Frame and Save; bisogna specificare il nome dell'icona, che deve essere uguale a quello del programma che rappresenta dal momento che IconEd aggiunge .info automaticamente. Fatto questo, l'icona è realizzata e deve essere collocata nella stessa directory del programma suo omonimo. A questo punto, basterà clickare doppiamente sull'icona per osservare un bellissimo Guru Meditation. Già, creare l'icona non è sufficiente; bisogna anche adattare il nostro programma per renderlo compatibile con lo Workbench.

Differenze tra CLI e Workbench

Il file che esce dall'assembler o meglio dal linker è direttamente eseguibile come comando CLI. All'attivazione il registro A0 contiene un puntatore alla stringa dei parametri digitati dopo il comando e D0 contiene la lunghezza di questa stringa; il CLI aggiunge sempre un linefeed al termine dei parametri e spesso risulta pratico utilizzarlo per trovare la fine della stringa invece che usare D0. Nel seguito si partirà dal presupposto che il programma che si vuol dotare di icona non necessiti di pa-

rametri (come quasi sempre avviene con i programmi richiamabili tramite Intuition) e non scrivano niente nella finestra del CLI, che usando lo Workbench non esiste. Quando si biclicka su di un'icona di tipo tool lo Workbench toglie il .info dal nome dell'icona e carica in memoria il file che ha quel nome, quindi gli invia un messaggio. Il sistema operativo dell'Amiga prevede infatti un protocollo ben preciso per lo scambio di messaggi e risposte tra i vari task; la gestione dei messaggi è affidata a Exec, che rappresenta il gradino più basso del software di sistema dell'Amiga (il più alto è lo Workbench) ed assume in questo ambito il ruolo del postino. Esistono delle entità chiamate porte che consentono il passaggio di messaggi e che assolvono la funzione di cassette postali. Il meccanismo funziona pressappoco così: quando un task vuole inviare un messaggio a un altro task deve in qualche modo localizzare la sua porta (è persino possibile cercarla per nome, dal momento che tutte le porte pubbliche ne possiedono uno); bisogna poi allocare una zona di memoria per contenere il messaggio, immagazzinarvelo e richiamare una routine di Exec (PutMsg) passandogli come parametri un puntatore al messaggio e uno alla porta di destinazione. Exec si incaricherà di inserire il messaggio nella lista dei messaggi appena arrivati che ogni porta possiede e invierà un segnale al task proprietario della porta stessa. Questo segnale può essere un vero e proprio interrupt generato via software o semplicemente un flag gestito da Exec che il task può andare a controllare di tanto in tanto richiamando un'apposita routine di Exec (Wait, WaitPort o GetMsg) per vedere se c'è posta per lui. Il destinatario preleva quindi il messaggio (o meglio un puntatore a quel messaggio, che in realtà non si è mai mosso di persona di un solo byte in tutto questo trambusto) con un'altra routine di Exec (GetMsg) e lo esamina a fondo. Quando il messaggio non serve più occorre confermare di averlo ricevuto e compreso chiamando un'ulteriore routine (ReplyMsg) che farà pervenire la risposta al mittente originario, che potrà finalmente deallocare la memoria usata per contenere il messaggio. Morale: i sistemi operativi multitasking sono più complessi di quanto sembri.

Un programma dopo essere stato lanciato in esecuzione da Workbench deve aspettare che gli venga inviato il messaggio iniziale e deve rispondere solo quando ha concluso il suo compito, perché ricevuta la risposta lo Workbench termina spietatamente il programma in questione e dealloca la memoria che gli era stata inizialmente riservata.

Un file di inclusione

Dai paragrafi precedenti emerge l'informazione che qualunque programma che non richieda parametri può essere adattato allo Workbench inserendovi all'inizio una serie di istruzioni che aspettano il messaggio e gli rispondono dopo aver richiamato il corpo centrale del programma. Questa sequenza di linee è funzionalmente indipendente dal programma centrale, può di conseguenza essere scritta una volta per tutte e memorizzata in un file chiamato per esempio IconHeader; quando in futuro verrà scritto un altro programma basterà inserire al suo inizio la singola linea INCLUDE "IconHeader" per renderlo Intuition-compatibile. Naturalmente IconHeader dovrà essere abbastanza furbo da accorgersi se il programma viene richiamato da Workbench o da CLI e regolarsi di conseguenza: non vogliamo infatti che il programma così modificato non possa più essere attivato da CLI. Esaminiamo il listato.

Prima di tutto bisogna procurarsi l'indirizzo base di Exec nella solita locazione 4. Il prossimo passo consiste nel trovare se stessi (sembra un problema filosofico), cioè localizzare la struttura di definizione del proprio task; la routine FindTask con argomento nullo posto in A1 porta a termine questo curioso compito. Passiamo il puntatore così ottenuto in A0 e usiamolo per controllare una certa variabile per sapere se il nostro processo è associato a un CLI; in caso affermativo bisogna saltare direttamente alla label WYYK317. "Chi è mai costui", griderà qualcuno inorridito; ebbene, quella label avrebbe anche potuto avere un qualunque altro nome ma in questo modo possiamo essere ragionevolmente certi di evitare spiacevoli collisioni con altre label definite nel programma principale. Se l'attivazione è stata eseguita dallo Workbench dobbiamo porre in A0 l'indirizzo della nostra porta e richiamare in sequenza WaitPort e GetMsg; fatto questo, salviamo sullo stack il puntatore al messaggio così ottenuto e richiamiamo il programma principale. Quando quest'ultimo ha ormai concluso il suo compito richiamiamo Forbid per bloccare temporaneamente il multitasking e infine ReplyMsg passando in A1 il puntatore al messaggio che avevamo salvato sullo stack. Infine rientriamo. L'unico punto oscuro dovrebbe essere la chiamata a Forbid: il fatto è che dopo aver risposto al messaggio lo Workbench ci dealloca la memoria senza tanti complimenti strappandoci letteralmente il tappeto da sotto i piedi prima di darci il tempo di rientrare, causando una confusione mostruosa. Forbid blocca ogni azione da parte dello Workbench fino al nostro rientro.

I programmi in Assembly pubblicati nei numeri precedenti che non richiedono parametri e non stampano niente nella finestra del CLI, come Cube e Lens, possono essere dotati di icona senza problemi.

```
*IconHeader by Paolo Russo - Amiga Magazine
            MOVE.L 4,A6
            SUB.L A1,A1
            JSR -294(A6)
            MOVE.L D0,A0
            TST.L 172(A0)
            BNE.S WYYK317
            LEA 92(A0),A0
            MOVE.L A0,A2
            JSR -384(A6)
            MOVE.L A2,A0
            JSR -372(A6)
            MOVE.L D0,-(A7)
            BSR.S WYYK317
            MOVE.L 4,A6
            JSR -132(A6)
            MOVE.L (A7)+,A1
            JSR -378(A6)
            MOVEQ #0,D0
            RTS
WYYK317     EQU *
```

LETTORE DI FILE .BMAP IN AMIGABASIC

Cosa ci sarà in questo .bmap file...?

di Fabio Biancotto

Mentre programmavo in AmigaBasic mi sono trovato di fronte a un problema. Provavo a lavorare con un programma terminale che mi desse la possibilità di richiamare una 'lista' o mi permettesse di generare una nuova CLI: dalla quale avrei fatto partire dei comandi DOS.

Mentro arrovistavo dentro i vari demo che sono inclusi nel disco Extras, ho notato un file che includeva una routine (tramite una chiamata ad a.library) che usava il comando DOS 'EXECUTE'. Ero sicuro che in quel file avrei trovato la risposta al mio problema.

Dopo aver sbirciato all'interno del programma, sono arrivato alla conclusione che, con l'appropiato file .bmap, potevo accedere alla maggior parte, se non a tutte, le funzioni costruite dall'Amiga. Il .bmap file chiamato dal mio programma, trovato nei demo dell'Extras, era il file dos.bmap (come "dos.library").

```
Initialize:
  FALSE = 0 : TRUE = -1
  BobRight = 230 : BobBottom = 90
  DEF FNArraySize& = 3 + INT((BobRight + 16)/16)*(BobBottom+1)*2
  DECLARE FUNCTION Move LIBRARY
  DECLARE FUNCTION SetDrMd LIBRARY
  CHDIR "BasicDemos"
  LIBRARY "graphics.library"
  CHDIR "/"
  Title$ = "   Lettore .bmap"
  WINDOW 1,Title$,(0,0)-(631,186),0
  PALETTE 0,0,0,0
  PALETTE 3,.8,.2,.1
  PALETTE 1,.1,.7,.1
  PALETTE 2,.9,.9,.1
  Rp& = WINDOW(8)
Start:
  COLOR 1,0 : CLOSE 1
  GOSUB NameRequestor
  IF NOT Okay THEN
    CLS : LIBRARY CLOSE
    WINDOW 1,"Lettore  bmap",(0,0)-(617,186),15,-1 : STOP
  END IF
  ON ERROR GOTO FileProb
  IF UCASE$(RIGHT$(FileName$,5)) <> ".BMAP" THEN
  FileName$ = FileName$ + ".bmap"
  END IF
  OPEN FileName$ FOR INPUT AS 1
  Prompt$ = "" :Prompt2$ = " Output su stampante?"
  GOSUB TNRequestor : IF Okay THEN fPrt = TRUE : GOTO Printer
  CLS
  LINE (3,13)-(628,170),1,b
  LINE (4,13)-(627,170),1,b
  Lenght& = LOF(1)
  LOCATE 23,1 : PRINT Lenght&;"Byte letti.    FILE: ";: COLOR 2,0 :
PRINT FileName$;
  CALL Move&(Rp&,10,10) : COLOR 3,0
  PRINT "Nome Routine           Indirizzo  d0 d1 d2 d3 d4 d5 d6 d7   a0
a1 a2 a3 a4"
  COLOR 1,0
GetTheFile:
  WHILE NOT EOF(1)
    FOR L = 3 TO 21
    IF EOF(1) THEN
      FOR J = L TO 21
        LOCATE J,2
        PRINT SPACE$(76)
      NEXT J
      GOTO Finished
    END IF
    GOSUB GetRoutName
      COLOR 1,0
      LOCATE L,2
      PRINT " ";Routine$
    GOSUB GetEntryAdd
      LOCATE L,28
      IF LEN(Address$(2)) = 1 THEN
        Address$(2) = "0" + Address$(2)
      END IF
      PRINT Address$(1);Address$(2)
      GOSUB GetRegInfo
        LOCATE L,35
        PRINT
      NEXT L
      COLOR 0,3
      LINE (386,173)-(612,185),3,bf
      LINE (386,174)-(610,184),0,b
      CALL Move&(Rp&,394,182) : PRINT "F1 continue <> F10 aborto";
      WaitKey:
        In$ = INKEY$ :IF In$ = "" THEN WaitKey
        IF In$ = CHR$(138) THEN
          LINE (386,173)-(612,185),0,bf
          GOTO Finished
        END IF
        IF In$ <> CHR$(129) THEN WaitKey
        COLOR 1,0
        LINE (386,173)-(612,185),0,bf
  WEND
  GOTO Finished

GetRoutName:
  Routine$ = ""
  GOSUB GetChar
  WHILE Char$ <> CHR$(0)
    Routine$ = Routine$ + Char$
    GOSUB GetChar
  WEND
  IF LEN(Routine$) < 30 THEN
    Routine$ = Routine$ + SPACE$(20 - LEN(Routine$))
  END IF
RETURN

GetEntryAdd:
  FOR ii = 1 TO 2
    GOSUB GetChar
    Address$(ii) = HEX$(ASC(Char$))
  NEXT ii
RETURN

GetRegInfo:
  LOCATE L,35 : PRINT SPACE$(42);
  WHILE Char$ <> CHR$(0)
    GOSUB GetChar
    COLOR 2,0
    Register = ASC(Char$)
    GOSUB R1
  WEND
  IF fPrt THEN PRINT #4," "
RETURN

GetChar:
  IF NOT EOF(1) THEN Char$ = INPUT$(1, 1)
RETURN

Finished:
  COLOR 3,0
  LOCATE 23,1
  PRINT SPACE$(78);
  LOCATE 23,1
  PRINT " Vuoi esaminare un altro  BMAP file (Y/N)?";
test:
  In$ = INKEY$ : IF In$ = "" THEN test
  IF UCASE$(In$) <> "Y" THEN
    CLS : CLOSE 1 : LIBRARY CLOSE : WINDOW CLOSE 1
    WINDOW 1,"BmapReader",(0,0)-(617,186),15,-1 : STOP
  END IF
  GOTO Start

R1:
  IF Register < 1 THEN RETURN
  IF Register > 8 THEN R2
  IF fPrt THEN
    PRINT #4,CHR$(141);TAB(32 + (Register*3));"#";
    RETURN
  END IF
  LOCATE L,(32 + (Register*3))
  PRINT "#"
  RETURN
```

```
R9:
  IF fPrt THEN
    PRINT #4,CHR$(141);TAB(34 + (Register*3));"#";
    RETURN
  END IF
  LOCATE L,(34 + (Register*3))
  PRINT "#"
RETURN

Printer:

  OPEN "LPT1:BIN" FOR OUTPUT AS 4
  PRINT #4,CHR$(14);"Contenuto del file ":FileName$
  PRINT #4," "
  WHILE NOT EOF(1)
    PRINT #4,"Nome Routine              Indirizzo  d0 d1 d2 d3 d4 d5 d6 d7
a0 a1 a2 a3 a4"
    PRINT #4,"------------              ---------  -- -- -- -- -- -- -- --
-- -- -- -- --"
    FOR L = 1 TO 54
    IF EOF(1) THEN
      GOTO Finished
    END IF
    GOSUB GetRoutName
      PRINT #4," ";Routine$;
    GOSUB GetEntryAdd
      IF LEN(Address$(2)) = 1 THEN
        Address$(2) = "0" + Address$(2)
      END IF
      PRINT #4,"  ";Address$(1);Address$(2)
    GOSUB GetRegInfo
    NEXT L
    PRINT #4,CHR$(12)
  WEND
  PRINT #4,CHR$(12)
  CLOSE 4 : fPrt = FALSE
  GOTO Start

FileProb:

  flag = ERR
  Prompt$ = ""
  Prompt2$ = " Errore! >>" + STR$(ERR)
  GOSUB YNRequestor
  IF NOT Okay THEN
    LIBRARY CLOSE
    CLOSE 1
    WINDOW CLOSE 1
    WINDOW 1,"LettoreSmap",(0,0)-(617,186),31,-1
    END
  END IF
  RESUME Start

NameRequestor:
  SizeG = FNArraySizeG\2
  DIM ScrSavG(SizeG)
  GET (40,40)-(230,90),ScrSavG
DrawRequestorToScreen2:
  LINE (40,40)-(230,90),1,bf
  'Box requestor principale
  LINE (40,40)-(230,90),0,b
  'Outline per il box requestor principale
  LINE (44,42)-(226,88),0,b
  'Seconda outline per box principale
  LINE (50,74)-(72,86),3,bf
  'Box del tasto OK
  LINE (50,74)-(72,86),0,b
  'Outline OK
  LINE (150,74)-(220,86),3,bf
  'Box del tasto CANCEL
  LINE (150,74)-(220,86),0,b
  'Outline CANCEL
  CALL MoveG(RpG,53,83)
  'Posizione per stampare il tasto OK
  COLOR 0,3 : PRINT "OK"
  CALL MoveG(RpG,160,83)
  'Posizione per stampare il tasto CANCEL
  COLOR 0,3 : PRINT "CANCEL"
  LINE (53,50)-(216,62),3,b
  Curs = 56 : LINE (Curs,52)-(Curs+7,60),2,bf
  'Stampa il pseudo-cursore
  CALL MoveG(RpG,53,71) : COLOR 0,1 : PRINT "Digita il nome file"
  C$ = INKEY$ : WHILE C$ <> "" : C$ = INKEY$ : WEND
  'Svuota il buffer della tastiera
  FileName$ = ""

AcceseLoop:
'Aspetto il click nel box stringa o nel CANCEL

  I = MOUSE(0) : X = MOUSE(1) : Y = MOUSE(2)
  IF I <> 0 THEN
    WHILE I <> 0 : I = MOUSE(0)   X = MOUSE(1) : Y = MOUSE(2) : WEND
    Y=Y-1
    IF X > 150 AND X < 220 AND Y > 74 AND Y < 86 THEN
    'Check per il CANCEL
      CALL SetOrMdG(RpG,2) : LINE(151,75)-(219,85),0,bf
      CALL SetOrMdG(RpG,1)
      Okay = FALSE : FOR Delay = 1 TO 1000 : NEXT Delay
      PUT (40,40),ScrSavG,PSET
      ERASE ScrSavG : COLOR 1,0 : RETURN
    END IF
    IF X > 53 AND X < 216 AND Y > 50 AND Y < 62 THEN
      LINE (Curs,52)-(Curs+7,60),0,bf
      FOR Delay = 1 TO 50 : NEXT Delay
      LINE (Curs,52)-(Curs+7,60),2,bf
      WHILE INKEY$ <> "": WEND
      GOTO Loop
    END IF
  END IF
GOTO AcceseLoop

Loop:
  C$ = INKEY$ : I = MOUSE(0)   X = MOUSE(1) : Y = MOUSE(2)
  IF I <> 0 THEN
    WHILE I <> 0 : I = MOUSE(0) : X = MOUSE(1) : Y = MOUSE(2) : WEND
    Y=Y-1
    IF X > 150 AND X < 220 AND Y > 74 AND Y < 86 THEN
    'Check per il CANCEL
      CALL SetOrMdG(RpG,2) : LINE (151,75)-(219,85),0,bf
      CALL SetOrMdG(RpG,1)
      Okay = FALSE : FOR Delay = 1 TO 1000 : NEXT Delay
      PUT (40,40),ScrSavG,PSET
      ERASE ScrSavG : COLOR 1,0 : RETURN
    END IF
    IF X > 50 AND X < 72 AND Y > 74 AND Y < 86 AND LEN(FileName$) > 0
THEN
     'Check per OK e lunghezza del file
      CALL SetOrMdG(RpG,2) : LINE (51,75)-(71,85),0,bf
      CALL SetOrMdG(RpG,1)
      FOR Delay = 1 TO 1000 : NEXT Delay : Okay = TRUE : fExist = TRUE
      PUT (40,40),ScrSavG,PSET
      ERASE ScrSavG : COLOR 1,0 : RETURN
    END IF
  END IF
  IF C$ = "" THEN GOTO Loop
  IF LEN(FileName$) = 0 THEN IF C$ < "A" AND ASC(C$) <> 13 GOTO Loop
  'Non permette di inserire un non-Alpha come primo carattere
  IF ASC(C$) = 13 THEN
    Okay = TRUE : fExist = TRUE
    PUT (40,40),ScrSavG,PSET
    ERASE ScrSavG : COLOR 1,0 : RETURN
  END IF
  IF ASC(C$) = 8 THEN
  'Cattura il BackSpace e fissa il display e il filename
    FileName$ = LEFT$(FileName$,LEN(FileName$)-1)
    LINE (Curs,52)-(Curs+7,60),1,bf
    Curs = Curs-8 : LINE (Curs,52)-(Curs+7,60),2,bf
    GOTO Loop
  END IF
  IF LEN(FileName$) = 19 THEN GOTO Loop
  IF ASC(C$) = 8 THEN Loop
  IF C$ < " " OR (C$ > "Z" AND C$ < "a") OR C$ > "z" GOTO Loop
  FileName$ = FileName$ + C$
  LINE (Curs,62)-(Curs+7,60),1,bf
  COLOR 0,1 : CALL MoveG(RpG,0,59) : PRINT PTAB(Curs);C$;
  Curs = Curs + 8 : LINE (Curs,52)-(Curs+7,60),2,bf
  GOTO Loop

YNRequestor:

  SizeG = FNArraySizeG\2
  'Per riservare memoria per il GET statement
  DIM ScrSavG(SizeG)
  GET (40,40)-(230,90),ScrSavG
  LINE (40,40)-(230,90),2,bf
  'Box requestor principale
  LINE (40,40)-(230,90),0,b
  LINE (44,42)-(226,88),0,b
  LINE (50,74)-(72,86),3,bf
  'Box del tasto OK
  LINE (50,74)-(72,86),0,b
  'Outline OK
  LINE (150,74)-(220,86),3,bf
  'Box del tasto CANCEL
  LINE (150,74)-(220,86),0,b
  'Outline CANCEL
  CALL MoveG(RpG,53,83)
  'Posizione del tasto OK
  COLOR 0,3 : PRINT "OK"
  CALL MoveG(RpG,160,83)
  'Posizione del tasto CANCEL
  COLOR 0,3 : PRINT "CANCEL"
  CALL MoveG(RpG,54,52)
  'Posizione per il primo testo
  COLOR 0,2 : PRINT Prompt$
  CALL MoveG(RpG,54,62)
  'Posizione per il secondo testo
  COLOR 3,2 : PRINT Prompt2$

GetButton2:
  'Aspetta che sia premuto il tasto di sinistra del mouse
  I = MOUSE(0) : X = MOUSE(1) : Y = MOUSE(2)
  IF I <> 0 THEN
    WHILE I <> 0 : I = MOUSE(0) : X = MOUSE(1) : Y = MOUSE(2) : Y=Y-1
    WEND
Cancel:
    'Check per vedere se il tasto CANCEL e' selezionato
    IF X > 150 AND X < 220 AND Y > 74 AND Y < 86 THEN
      CALL SetOrMdG(RpG,2)
      LINE (151,75)-(219,85),0,bf
      CALL SetOrMdG(RpG,1)
      FOR Delay = 1 TO 1000 : NEXT Delay
      PUT (40,40),ScrSavG,PSET
      ERASE ScrSavG
      'Cancella l'Array
      Okay = FALSE
      COLOR 1,0
      RETURN
    END IF
Ok:
   'Check per vedere se il tasto OK e' selezionato
    IF X > 50 AND X < 72 AND Y > 74 AND Y < 86 THEN
      CALL SetOrMdG(RpG,2)
      LINE (51,75)-(71,85),0,bf
      CALL SetOrMdG(RpG,1)
      FOR Delay = 1 TO 1000 : NEXT Delay
      PUT (40,40),ScrSavG,PSET
      Okay = TRUE
      ERASE ScrSavG
      'Cancella l'Array
      COLOR 1,0
      RETURN
    END IF
  END IF
  GOTO GetButton2
  'Fino a quando il tasto e' selezionato
```

Dopo essermi studiato il programma sembrava che questo interagisse con la library; andai allora velocemente a riferirmi alla spiegazione dei .bmap file nel manuale. Questo mi fece emergere ancora più domande; e ancora, se avessi potuto entrare nel .bmap file sarei stato capace di fare buon uso di ciò che avrei trovato nel suo interno?

Ho usato il FileZap che ho lanciato vicino all'area BBS e ho trovato che c'erano molte funzioni nel file dos.bmap che contenevano solo 'EXECUTÈ. Sono partito alla ricerca nei meandri dei miei "developer manual", e ho trovato che la library chiamata nel .bmap file era direttamente esclusa dai manuali AmigaDos.

Dopo aver confrontato i dati che ho visto usando il FileZap con il formato che il manuale Basic diceva di dover possedere, mi

ero totalmente perso. Con l'eccezione del nome delle routine, il file apparso conteneva gli inutili scarti che erano supposti per rappresentare il registro usato dalla routine.

Come in molti programmi da me scritti, decisi che dovevano esserci molti altri modi per esaminare il file e determinarne quindi le parti utilizzabili.

Pertanto il "lettore Bmap" viene concepito per esaminare i file .bmap che compaiono nelle piu' svariate stazioni BBS della nazione.

Note al programma

Quando carichiamo e lanciamo il lettore Bmap, questo guarderà nella directory di default alla ricerca del file graphics.bmap. Aprirà la library e userà due delle routine che sono a disposizione. Solo la routine Move() e la routine SetDrMd(). La routine Move() ci permette di specificare l'indirizzo di un pixel, piuttosto che la cella di un carattere, quando posizioniamo del testo sullo schermo. Il comando CALL è utilizzato con la sintassi che segue:

```
CALL Move&(rastPort&,LocazioneX,
     LocazioneY)
```

Possiamo prendere il valore del puntatore rastPort dalla funzione WINDOW(8). Nel programma del lettore, al puntatore è assegnata la variabile Rp&.

Questa variabile serve a chiamare la funzione WINDOW(8) dopo che abbiamo definito la struttura della finestra. Nel caso che si abbia bisogno di lavorare con più di una finestra, setteremo alle variabili lo stesso numero delle finestre (per esempio: WINDOW 1 = Rp1&, WINDOW 2= Rp2&, eccetera).

Un importante fattore da tener presente quando si usa Move(), è l'indirizzo della rastPort: è l'angolo superiore sinistro della finestra e non tiene conto dei bordi o la della title/drag bar. Perciò, quando posizioniamo qualcosa nello schermo, dobbiamo aggiungere undici nelle coordinate Y e uno nelle coordinate X in modo di ottenere le stesse coordinate X e Y usate dagli altri comandi grafici del BASIC (LINE, CIRCLE, GET, PUT, e altri).

Un altro importante punto che possiamo aver notato, quando abbiamo usato nel programma DECLARE FUNCTION SetDrMd LIBRARY, è che abbiamo dichiarato la funzione come una variabile SHORT e poi l'abbiamo chiamata come LONG(SetDrMd&()). Il nome attuale non la specifica necessariamente come LONG perché sta solo cercando il nome della funzione nel .bmap file. Infatti dichiarando la funzione come LONG, hò visto causare una GURU! Questo non succede tutte le volte, ma perché aiutare un "disastro"?

La routine SetDrMd() ci permette di accedere nella library ai vari modi di scrittura. Questi modi sono JAM1, JAM2, COMPLEMENT e INVERSE. Quando scriviamo normalmente sullo schermo siamo in modo JAM2. Non entrerò nei dettagli circa le differenze di questi modi in questa sede, però sono tutti a disposizione usando la routine SetDrMd(). La sintassi è:

```
CALL SetDrMd&(Rp&,Mode%)
```

Dove Rp& è, lo stesso, indirizzo del puntatore rastPort e il Mode% è il valore del modo scrittura che puoi usare (0,1,2,4 o altre combinazioni di bitwise).

Queste funzioni sono usate per collocare un testo in una non- LOCATE posizione per avere dei display ancora più originali. Inoltre, usiamo SetDrMd() quando ci serve la selezione dei tasti Requestor.

La routine Requestor è presentata separatamente all'articolo e include il completo commento del lavoro e delle variabili da assegnare ad altri programmi per dotarli di questa routine.

Uso del lettore di bmap

Dopo aver lanciato, il lettore bmap apparirà il Requestor e ci chiederà il nome del file da cercare nella corrente directory o in una specificata. Quando si digita il nome del file non è necessario aggiungere il suffisso .bmap, sarà il programma ad aggiungerlo.

Se il file non viene trovato il Requestor riapparirà informandoci, e richiederà di poter cercare un altro file o di uscire.

Se il file è allocato, il programma visualizza 20 linee di informazioni inerenti a quanto ha scoperto nel file.

Le informazioni visualizzate sono: Il nome della routine Il vettore offset di salto d'indirizzo Il registro usato dalla funzione (contrassegnato da #).

Credo che questo programma non verrà utilizzato molto frequentemente dato il suo specifico uso e le sue limitazioni ma è molto utile per poter comprendere l'uso specifico delle chiamate alle funzioni delle library e aiutare i programmatori a usarle nei loro programmi.

Struttura BHMD		Struttura GRAB		Struttura CRNG	
UWORD	w,h	WORD	x,y	WORD	pad1
WORD	x,y			WORD	rate
UBYTE	nPlanes	Struttura DEST		WORD	active
UBYTE	Masking	UBYTE	depth	UBYTE	low, high
UBYTE	Compression	UBYTE	pad1		
UBYTE	pad1	UWORD	PlanePick	Struttura CORT	
UWORD	transparentColor	UWORD	PlaneOnOff	WORD	direction
UBYTE	xAspext,yAspect	UWORD	PlaneMask	UBYTE	start,end
WORD	pageWidth, pageHeight			LONG	seconds
Struttura	SPRT	Struttura CAMG		LONG	microseconds
UWORD	precedence	LONG	ViewMode	WORD	pad

Tabella 1: alcune proprietà dell'ILBM

gative tutte le altre. Vedremo nel seguito cosa tutto ciò significhi esattamente.

Fondamentale è il concetto di hunk ID; ogni volta che si introduce un blocco di dati bisogna contrassegnarlo con una long word costituita da quattro codici ASCII corrispondenti a lettere maiuscole o cifre. Per esempio ogni file IFF deve iniziare con "FORM", "LIST" o "CAT ". L'ID è poi generalmente seguito da una long word contenente la lunghezza di tutto il resto del blocco. Attenzione però che quest'ultima può essere dispari, nel qual caso in base al criterio della lunghezza sempre pari ci sarà un byte vuoto prima del prossimo blocco per riottenere l'allineamento a indirizzo pari. L'estrema utilità della long word contenente la lunghezza del blocco consiste nella possibilità offerta al programma caricatore di saltare completamente i blocchi con ID sconosciuto. Per esempio Deluxe Paint salva ogni schermata includendo quattro blocchi non standard CRNG (color range) che descrivono i quattro range di colori disponibili, ma ogni programma grafico che non conosce questi blocchi può saltarli senza danno e leggere solo la pura immagine.

L'IFF è strutturato. Gli ID LIST e CAT introducono gruppi di oggetti ognuno dei quali può essere anche un altro LIST o CAT e ciò consente la realizzazione di strutture di dati molto complesse. Gli oggetti singoli sono invece introdotti dalla parola FORM e nel seguito si partirà dal presupposto di dover salvare o caricare una schermata singola. Ogni blocco può contenere dei sottoblocchi (chiamati proprietà), la presenza di alcuni dei quali può essere facoltativa.

Il formato ILBM

Il più importante tipo di blocco per la rappresentazione di informazioni grafiche si chiama ILBM da Interleaved BitMap e i suoi sottoblocchi opzionali sono BMHD, CMAP, GRAB, DEST, SPRT, CAMG, BODY, CRNG e CCRT dei quali il primo è anche l'unico la cui presenza sia obbligatoria. Un file destinato a contenere un'immagine singola dovrà iniziare con la parola FORM seguita dalla lunghezza del resto del file e dalla parola ILBM. A questa intestazione iniziale seguiranno tutti gli altri blocchi, ognuno dei quali inizierà con l'ID del blocco e con una long word contenente la lunghezza in byte del resto del blocco, continuerà con il corpo del blocco stesso e, se la lunghezza era dispari, finirà con un byte contenente zero per riottenere un indirizzo pari; naturalmente se la lunghezza era già pari questo ulteriore byte non sarà inserito. Nel seguito per struttura di un blocco si intenderà il corpo vero e proprio del blocco stesso, senza menzionare l'ID e la lunghezza che lo precedono.

Il blocco BMHD (BitMap Header) dovrebbe essere posto per primo, anche se ciò non è obbligatorio, poiché contiene molte informazioni importanti, ed è visibile in tabella 1. I primi due parametri (UWORD significa unsigned word, ovvero numero a 16 bit privo di segno) indicano la larghezza e l'altezza dell'immagine, mentre i successivi due indicano una posizione preferenziale per l'inserzione in altre immagini (solitamente sono posti a zero). Il prossimo parametro indica il numero di piani di bit e quindi indirettamente il numero di colori, segue poi il tipo di mascheratura (importante per un pennello, più che per un'immagine generica) che può assumere quattro valori:

zero indica l'assenza di qualunque maschera;

uno significa che la maschera esiste sotto forma di un ulteriore piano di bit, nel quale i bit posti a zero indicano un pixel da reputarsi trasparente;

due significa che uno dei colori, per l'esattezza quello indicato dalla variabile transparentColor presente poco più avanti nella struttura, è in realtà da considerarsi trasparente;

tre indica la possibilità di costruire una maschera basandosi sui bordi dell'oggetto rappresentato nell'immagine.

Il parametro successivo indica se la schermata è stata compressa (uno) o no (zero); l'algoritmo di decompressione standard prevede che, chiamando N il valore del primo byte dell'immagine che si incontra nel blocco BODY, se N è positivo i successivi N+1 byte devono essere copiati letteralmente mentre se è negativo il byte successivo deve essere replicato -N+1 volte. La procedura viene ripetuta fino alla fine della prima riga del primo bitplane e viene poi ripetuta per gli altri bitplane e per tutte le altre righe. Deluxe Paint comprime sempre le sue immagini e questo semplice fatto obbliga ogni caricatore IFF degno di questo nome a supportare l'algoritmo di espansione. Segue, nella struttura BMHD, un byte inutilizzato (a scopo di allineamento), la variabile transparentColor che è stata già descritta, una coppia di byte che indicano il rapporto x:y di un singolo pixel (informazione diretta evidentemente alle stampanti) e infine le dimensioni della pagina con la quale visualizzare l'immagine (la pagina è intesa come una finestra sull'immagine, ma normalmente queste dimensioni dovrebbero coincidere).

Il blocco CMAP (Color Map) ha una struttura molto più semplice, essendo una sequenza di triplette di byte contenenti rispettivamente le componenti di rosso, di verde e di blu presenti in ogni colore. In realtà l'Amiga possiede solo 16 livelli di rosso, verde e blu, non 256, e basterebbero solo quattro bit per ogni componente; lo standard prevede che in questi casi si debba utilizzare la parte più significativa di ogni byte lasciando il resto a zero.

Il blocco GRAB è costituito da una coppia di coordinate che localizzano un punto particolare dell'immagine (da usarsi per gli sprite e i pennelli).

Il blocco DEST illustra come disporre i piani di bit presenti nel file in un'immagine che ne possiede un numero maggiore. Il primo campo contiene il numero di piani di bit presenti nel file, mentre il secondo serve solo a scopo di allineamento. Ogni bit in uno dei parametri successivi corrisponde a un

bitplane di destinazione (il bit#0 è associato al primo bitplane e gli altri seguono in ordine); PlanePick fornisce la corrispondenza tra i bitplane sorgente e quelli destinazione, PlaneOnOff indica quale valore assegnare globalmente ai rimanenti piani di destinazione e PlaneMask indica quali piani di destinazione devono essere realmente alterati e quali no. Supponiamo di avere tre bitplane e di voler costruire con questi un'immagine destinazione con sei piani, dei quali il primo, il secondo e il quinto uguali rispettivamente al primo, secondo e terzo dei piani sorgente mentre il terzo dovrebbe essere riempito di uni, il quarto di zeri e il sesto dovrebbe restare inalterato; i tre parametri varrebbero rispettivamente $010011, $000100 e $011111. Il blocco DEST è quindi di difficile gestione, ma molto raro ed è previsto che i caricatori più semplici possano ignorarlo.

Il blocco SPRT contiene una singola word, la priorità dello sprite (supponendo che l'immagine appartenga a uno sprite).

Il blocco CAMG (Commodore Amiga) contiene una long word i cui 16 bit inferiori contengono quei flag normalmente presenti nel campo Mode di un ViewPort, cioè informazioni circa il modo grafico (HAM, dual playfield, Hires, Lace..).

Il blocco BODY è il più importante e deve essere posto per ultimo nel file, perché solitamente i caricatori si fermano dopo averlo incontrato e gestito. Questo blocco contiene tutta l'immagine, riga per riga, in modo interleaved, ossia si parte considerando la prima riga del primo bitplane, poi la stessa riga del secondo bitplane e così via fino all'ultimo bitplane e opzionalmente al bitplane di mascheratura, poi si passa alla seconda riga e così via fino in fondo. Se l'immagine è compressa lo standard assicura che la compressione si ferma alla fine di ogni riga. Inoltre, ogni riga (non compressa) di un'immagine deve avere un numero pari di byte.

I blocchi non standard CRNG e CCRT (rispettivamente usati da Deluxe Paint e GraphiCraft) descrivono un range di colori e informazioni per alternarli in tempo reale.

Il programma View

Il programma presente in questo articolo consente di caricare da CLI un'immagine che viene visualizzata in uno schermo di Intuition dotato di gadget invisibili per spostarlo in alto, in basso o dietro altri schermi e per chiuderlo. Il pathname della schermata deve essere digitato dopo la parola View. È possibile caricare molte immagini simultaneamente con il comando run:

run View immagine1
run View immagine2
run View immagine3

In questo modo tutti i View girano in multitasking contemporaneamente, ognuno mostrando un'immagine diversa. L'unico limite è la quantità di chip RAM disponibile. Il programma ignora tutti i blocchi eccetto BMHD, CMAP, CAMG e naturalmente BODY e parte dal presupposto che non esista alcuna maschera; non è inoltre particolarmente protetto e se il file contiene errori potrebbe anche saltare.

```
* View by Paolo Russo - Amiga Magazine
VIEW        MOVE.L A0,A5
            MOVE.L 4,A6
            LEA LIBRARIES(PC),A2
            LEA EXEC(PC),A3
            MOVE.L A6,(A3)+
            MOVEQ #2,D2
L1          MOVEQ #0,D0
            MOVE.L A2,A1
            JSR -552(A6) ; D0=OpenLibrary(A1,D0)
            MOVE.L D0,(A3)+
L2          TST.B (A2)+
            BNE.S L2
            DBRA D2,L1
            MOVE.L #1024,D0
            MOVEQ #3,D1
            JSR -196(A6) ; D0=AllocMem(D0,D1)
            MOVE.L D0,(A3)+
            MOVE.L D0,A2
            MOVE.L D0,A1
            MOVE.B (A5)+,D0
L3          MOVE.B D0,(A1)+
            MOVE.B (A5)+,D0
            CMP.B #10,D0
            BNE.S L3
            CLR.B (A1)
            MOVE.L A2,D1
            MOVE.L DOS(PC),A6
            MOVE.L #1005,D2
            JSR -30(A6) ; D0=Open(D1,D2)
            MOVE.L D0,D7
            BEQ ERR1
            MOVEQ #12,D3
            BSR.S READ
            CMP.L #'FORM',(A1)+
            BNE ERR2
            CMP.L #'ILBM',4(A1)
            BNE ERR2
LOADLOOP    MOVEQ #8,D3
            BSR.S READ
            MOVE.L (A1)+,D0
            MOVE.L (A1)+,D3
            ADDQ.L #1,D3
            AND.B #$FE,D3
            LEA PROPS(PC),A0
            MOVEQ #2,D1
L4          MOVE.L (A0)+,A3
            CMP.L (A0)+,D0
            DBEQ D1,L4
            BNE.S L6
            BSR.S READ
            MOVE.L D3,(A3)+
L5          MOVE.B (A1)+,(A3)+
            SUBQ.W #1,D3
            BNE.S L5
            BRA.S LOADLOOP
L6          CMP.L #'BODY',D0
            BEQ.S LOADSCREEN
            MOVE.L D7,D1
            MOVE.L D3,D2
            MOVEQ #0,D3
            JSR -66(A6) ; D0=Seek(D1,D2,D3)
            TST.L D0
            BMI ERR2
            BRA.S LOADLOOP
READ        MOVE.L D7,D1
            MOVE.L A2,D2
            JSR -42(A6) ; D0=Read(D1,D2,D3)
            MOVE.L (A7)+,A0
            CMP.L D0,D3
            BNE ERR2
            MOVE.L A2,A1
            JMP (A0)
LOADSCREEN  MOVEM.W LX(PC),D1-D5
            LSR.W #8,D5
            LEA NEWSCREEN(PC),A0
            LEA NEWWINDOW(PC),A2
            MOVEM.W D1-D2/D5,4(A0)
            MOVEQ #0,D3
            CMP.W #500,D1
            BCS.S LORES
HIRES       OR.W #$8000,D3
LORES       CMP.W #330,D2
            BCS.S NOLACE
LACE        OR.W #4,D3
NOLACE      MOVE.L CAMG(PC),D0
            BEQ.S NOCAMG
            MOVE.L VIEWMODE(PC),D3
NOCAMG      MOVE.W D3,12(A0)
            MOVE.L INTUITION(PC),A6
            JSR -198(A6) ; D0=OpenScreen(A0)
            MOVE.L D0,30(A2)
            MOVE.L D0,A3
            BEQ ERR2
            MOVE.L A2,A0
            JSR -204(A6) ; D0=OpenWindow(A0)
```

```
            MOVE.L D0,A2
            LEA CMAP(PC),A4
            MOVE.L (A4)+,D4
            BEQ.S NOCMAP
            MOVE.L GRAPHICS(PC),A6
            DIVU #3,D4
            CMP.W #32,D4
            BLS.S L6
            MOVEQ #32,D4
L6          MOVEQ #0,D6
            BRA.S L7
SETCOLOR    LEA 44(A3),A0
            MOVE.W D6,D0
            MOVEQ #0,D1
            MOVE.B (A4)+,D1
            LSR.B #4,D1
            MOVEQ #0,D2
            MOVE.B (A4)+,D2
            LSR.B #4,D2
            MOVEQ #0,D3
            MOVE.B (A4)+,D3
            LSR.B #4,D3
            JSR -288(A6) ; SetRGB4(A0,D0,D1,D2,D3)
            ADDQ.W #1,D6
L7          DBRA D4,SETCOLOR
NOCMAP      MOVE.L DOS(PC),A6
            MOVEM.L 192(A3),D0-D4/A0
            MOVEM.L D0-D4/A0,BITPLANES
            MOVE.B COMPRESS(PC),D0
            BNE.S COMP1
COMP0       MOVE.W 186(A3),D6
C1          LEA BITPLANES(PC),A4
            MOVE.W D5,D4
C2          MOVE.L D7,D1
            MOVE.L (A4),D2
            MOVEQ #0,D3
            MOVE.W 184(A3),D3
            JSR -42(A6) ; D0=Read(D1,D2,D3)
            ADD.L D3,(A4)+
            SUBQ.W #1,D4
            BNE.S C2
            SUBQ.W #1,D6
            BNE.S C1
            BRA.S LOADED
COMP1       MOVEQ #0,D3
            MOVE.W 186(A3),D6
C3          LEA BITPLANES(PC),A4
            MOVE.W D5,D4
C4          MOVE.L (A4),A0
            MOVE.W 184(A3),D2
C5          BSR.S GETBYTE
            EXT.W D0
            BMI.S REPLICATE
LITERAL     SUB.W D0,D2
            BSR.S GETMANY
            BRA.S NEXT
REPLICATE   NEG.W D0
            MOVE.W D0,D1
            BSR.S GETBYTE
            SUB.W D1,D2
C7          MOVE.B D0,(A0)+
            DBRA D1,C7
NEXT        SUBQ.W #1,D2
            BHI.S C5
            MOVE.L A0,(A4)+
            SUBQ.W #1,D4
            BNE.S C4
            SUBQ.W #1,D6
            BNE.S C3
            BRA.S LOADED
GETBYTE     MOVE.B (A1)+,D0
            DBRA D3,RET
            BSR.S FILLBUFFER
            BRA.S GETBYTE
GETMANY     MOVE.B (A1)+,(A0)+
            DBRA D3,OK
            SUBQ.L #1,A0
            BSR.S FILLBUFFER
            BRA.S GETMANY
```

```
OK          DBRA D0,GETMANY
RET         RTS
FILLBUFFER  MOVE.L #1024,D3
            MOVEM.L D0-D2/A0,-(A7)
            MOVE.L D7,D1
            MOVE.L BUFFER(PC),D2
            JSR -42(A6) ; D0=Read(D1,D2,D3)
            MOVE.L D2,A1
            MOVEM.L (A7)+,D0-D2/A0
            RTS
LOADED      MOVE.L EXEC(PC),A6
            MOVE.L 86(A2),A0
            MOVE.B 15(A0),D1
            MOVEQ #0,D0
            BSET D1,D0
            JSR -318(A6) ; D0=Wait(D0)
            MOVE.L INTUITION(PC),A6
            MOVE.L A2,A0
            JSR -72(A6) ; CloseWindow(A0)
            MOVE.L A3,A0
            move.w #0,D
            JSR -66(A6) ; CloseScreen(A0)
ERR2        MOVE.L DOS(PC),A6
            MOVE.L D7,D1
            JSR -36(A6) ; Close(D1)
ERR1        MOVE.L EXEC(PC),A6
            MOVE.L BUFFER(PC),A1
            MOVE.L #1024,D0
            JSR -210(A6) ; FreeMem(A1,D0)
            LEA DOS(PC),A2
            MOVEQ #2,D2
C8          MOVE.L (A2)+,A1
            JSR -414(A6) ; CloseLibrary(A1)
            DBRA D2,C8
            MOVEQ #0,D0
            RTS
EXEC        DC.L 0
DOS         DC.L 0
INTUITION   DC.L 0
GRAPHICS    DC.L 0
BUFFER      DC.L 0
BITPLANES   DC.L 0,0,0,0,0,0
NEWSCREEN   DC.W 0,0,0,0,0,0
            DC.W 0,15
            DC.L 0,0,0,0
NEWWINDOW   DC.W 0,0,50,50,0
            DC.L $200,$20846
            DC.L 0,0,0,0,0
            DC.W 0,0,0,0,15
PROPS       DC.L CAMG,'CAMG'
            DC.L BMHD,'BMHD'
            DC.L CMAP,'CMAP'
CAMG        DC.L 0
VIEWMODE    DC.L 0
BMHD        DC.L 0
LX          DC.W 0
LY          DC.W 0
X           DC.W 0
Y           DC.W 0
NPLANES     DC.B 0
MASKING     DC.B 0
COMPRESS    DC.B 0
PAD1        DC.B 0
TRANSPCOL   DC.W 0
XASPECT     DC.B 0
YASPECT     DC.B 0
PAGEWIDTH   DC.W 0
PAHEHEIGHT  DC.W 0
CMAP        DC.L 0
            DC.L 0,0,0,0,0,0,0,0
            DC.L 0,0,0,0,0,0,0,0
            DC.L 0,0,0,0,0,0,0,0
            DC.L 0,0,0,0,0,0,0,0
            DC.L 0,0,0,0,0,0,0,0
            DC.L 0,0,0,0,0,0,0,0
LIBRARIES   DC.B 'dos.library',0
            DC.B 'intuition.library',0
            DC.B 'graphics.library',0
            END
```

KINDWORDS

di Giorgio Dose

Un word processor in italiano

Finalmente qualcosa si sta muovendo nel mondo del software italiano dedicato all'Amiga ed anche se i primi programmi non sono altro che delle traduzioni delle corrispondenti versioni inglesi si incomincia a respirare un po' d'aria nuova e a intravedere la possibilitàdi utilizzo di questi programmi anche in ambienti professionali. È un peccato infatti che una macchina versatile come l'Amiga sia destinata, a parte qualche caso sporadico, a rimanere relegata tra le mura domestiche di hobbysti ed appassionati.

Il programma che andiamo a presentare è KindWords, un wordprocessor prodotto dalla The Disc Company, tradotto e pubblicato in Italia dalla C.T.O. di Bologna e commercializzato dalla stessa con il marchio Commodore Software by C.T.O.

Confezione

KindWords si presenta in un cofanetto di robusta plastica contenente due dischetti, un manuale di una trentina di pagine ed una cartolina che, inviata alla CTO, consente di ricevere un aggiornamento sui nuovi prodotti ed un'assistenza telefonica gratuita per 30 giorni prolungabile fino a 12 mesi con l'invio dell'apposito tagliando fornito dalla CTO stessa.

Dei dischetti, uno contiene il programma vero e proprio mentre l'altro, denominato Superfont, racchiude dei set di caratteri particolari che consentono una elevata qualità di stampa anche con stampanti economiche.

Il manuale è piuttosto conciso ma esauriente anche se presuppone una certa familiarità con l'uso dei wordprocessor e delle loro funzioni più comuni. Ma con Amiga tutto risulta facilitato grazie ai suoi menu e alle sue finestre.

I due dischetti non sono protetti e possono venir duplicati, per avere una copia di scorta, con molta facilità. È pure possibile l'installazione del programma su disco rigido.

Tutte le operazioni principali vengono svolte tramite l'interfaccia Intuition di Amiga, cioè menu a discesa e finestre, tuttavia la maggior parte di questi comandi sono accessibili anche da tastiera e possono tornare utili alle persone che hanno acquistato una certa familiarità con KindWords e non vogliono perdere neanche il tempo necessario per accedere ai menu

La filosofia di KindWords

KindWords è un programma molto potente che, oltre a possedere le tradizionali caratteristiche di un wordprocessor, consente l'accesso a molte funzioni avanzate quali l'inserimento di immagini e la loro manipolazione, l'unione tra i vari file, la scelta tra diversi tipi di caratteri e stili compresi apici, pedici e simboli, la possibilità delle intestazioni e dei piè di pagina oltre alla ricerca e sostituzione di testo e al mail-merge ovvero alla possibilità di creare molte versioni personalizzate di un unico documento. Un esempio tipico è la stesura di una stessa lettera che deve essere inviata a più indirizzi. Con il comando mail-merge è possibile inserire, in qualsiasi punto del testo, delle parole identificate in modo particolare che, in fase di stampa, verranno sostituite con le informazioni contenute in un file separato detto "lista di stampa".

Quello che appare sullo schermo è ciò che verrà stampato (WYSIWYG) e quindi i vari tipi di carattere ed i vari stili (grassetto, corsivo, sottolineato) compaiono sullo schermo come appariranno in stampa. Questo consente di avere un'idea precisa di quello che sarà il documento finale.

Con questo programma è bene definire, fin dall'inizio, l'impostazione della pagina con i suoi margini, l'interlinea, i caratteri eccetera, anche se è sempre possibile apportarvi qualunque modifica con estrema semplicità.

KW (chiamiamolo così confidenzialmente) funziona anche con soli 512 K di memoria ma esistono alcune restrizioni:
— non lanciare il programma da CLI
— chiudere tutte le finestre CLI
— usare uno schermo a quattro colori
— non usare le Preferences mentre KW è aperto
— non usare nessun drive esterno
— usare meno combinazioni di tipi e stili di carattere stampando con Superfonts
— inserire immagini con pochi colori.

Per utilizzare al meglio KW è consigliabile un'espansione della memoria ad 1 Mb per coloro che ancora non la possedessero.

Il programma legge la configurazione delle Preferences quando viene aperto. Per cambiare le Preferences mentre KW è attivo e per utilizzare la nuova configurazione è necessario chiudere KW e poi riaprirlo.

Prova su strada

La finestra di KW è costituita dalle seguenti parti:

— in alto una riga dove compare il titolo ed il numero della pagina; premendo il pulsante destro del mouse compaiono i vari menu;

— più in basso il righello suddiviso in numero caratteri sul quale trovano posto gli indicatori di margine destro, sinistro e di rientro del paragrafo; essi possono venir spostati facilmente usando il mouse e trascinati lungo il righello nella posizione voluta;

— sotto al righello ammiccano le icone della tabulazione normale e decimale (per l'allineamento dei numeri), dell'interlinea, semplice o doppia, e dell' allineamento del testo, a destra, a sinistra, al centro o giustificato.

— quindi c'è la finestra a pieno schermo per lo sviluppo del documento che contiene fino ad un massimo di 200 caratteri per linea. Detta finestra è fornita dei soliti gadgets di chiusura, di modifica delle dimensioni e di spostamento davanti o dietro ad altre finestre eventualmente aperte ed è dotata di sbarre di scorrimento orizzontale e verticale per rapidi spostamenti lungo il documento.

Scrivere con KW è veramente molto piacevole; fin dall'inizio si ha l'impressione di una eccezionale comodità d'uso unita a delle prestazioni notevoli. Il cursore si muove rapidamente lungo il testo e non presenta antipatici ritardi come in altri programmi del genere.

Lo spostamento del cursore, come in ogni wordprocessor che si rispetti, avviene in diversi modi. È possibile usare i tasti con le frecce, da soli o combinati con i soliti Ctrl, Alt, Shift e Amiga per gli spostamenti rapidi all'inizio e fine del testo, della pagina, del paragrafo o della riga, ma sono consentiti anche spostamenti più complessi con l'uso dei tasti Alt o Ctrl e il paddle numerico. Molto comodo è lo spostamento all'inizio di una determinata pagina che si ottiene mantenendo premuto il tasto Ctrl mentre si digita il numero della pagina desiderata.

Gli spostamenti del cursore possono naturalmente avvenire anche tramite il mouse; è sufficiente spostare il puntatore in un qualsiasi punto del testo, clickare sul pulsante sinistro ed in un batter d'occhio (anche se ormai questo movimento appare un po' lentino) il cursore apparirà nella posizione voluta.

Se il documento è piu grande dello schermo di KW, quando il cursore viene spostato verso un margine fissato fuori dallo schermo, il programma fa scrollare automaticamente il testo nella direzione voluta.

Molto funzionale risulta anche la possibilità di cancellare la parola corrente, o quella a destra o a sinistra del cursore, con l'uso combinato dei tasti Alt o Shift con Del o Backspace, senza dover evidenziare la parola stessa e operare con i menu.

Anche la selezione delle parole, righe o blocchi di testo per apportarvi delle modifiche è molto rapida. È sufficiente infatti scorrere con il mouse sopra al testo con il tasto sinistro premuto. La parte evidenziata verrà colorata in celeste.

Esistono comunque dei modi ancora più veloci: per selezionare una parola clickare due volte il mouse con il puntatore in qualsiasi punto della parola stessa, per selezionare una frase clickare invece tre volte su una qualsiasi delle parole comprese nella frase interessata.

Pure in questo caso, per un maggior comfort, esiste la possibilità di usare alcuni tasti che, per le operazioni di cui sopra, corrispondono ai primi cinque tasti funzione.

Dopo aver selezionato una parte di testo risulta facilissimo, tramite le opzioni dei menu, effettuare sullo stesso tutte le possibili variazioni: modificare il font, lo stile, i margini, la giustificazione, l'interlinea e così via e questo senza alterare il resto del testo. KW accetta infatti, all'interno di uno stesso documento, paragrafi, frasi o anche solo parole scritte con diversi font o formati.

Dopo un'operazione di cancellatura è possibile, in caso di errore, ritornare indietro usando l'opzione Disfa Azzeramento del menu Modifica a patto che il cursore non sia stato ancora mosso.

Per sostituire del testo non è necessario prima cancellare quello che non serve, è sufficiente selezionare la parte da buttare con uno dei vari metodi visti in precedenza e poi cominciare ad introdurre il nuovo testo. Questo provocherà automaticamente l'annullamento di quanto evidenziato. Anche in questo caso è possibile ritornare alla situazione di partenza

selezionando l'opzione Disfa Introduzione del menu Modifica. La funzione però opera solamente sugli ultimi 256 caratteri introdotti.

Tutte le impostazioni di un documento (margini, tabulazioni, eccetera) possono venir selezionate anche attraverso un'unica finestra che appare con l'opzione Formato del menu omonimo.

Inserimenti

Il programma consente di inserire un file in qualsiasi punto del documento corrente. Selezionando l'opzione Inserisci del menu File si aprirà una finestra attraverso la quale viene prelevato da disco il nuovo file; esso, alla fine dell'operazione comparirà, nel documento, di seguito al cursore. Allo stesso modo possono essere inserite delle immagini purché generate nel formato Interleaved File Format (IFF). Le immagini possono avere qualsiasi risoluzione ma se sono interlacciate KW convertirà le stesse in modo da visualizzarle senza le linee eccedenti. Il disegno inoltre deve contenere al massimo 16 colori; se dovesse possederne di più essi verranno ricalcolati da KW in modo da utilizzarne solamente 16. Ogni immagine viene conservata im memoria con i propri colori originali (max 16) ma è visualizzata sullo schermo con il numero di colori che KW sta usando in quel momento. Essi normalmente sono 4 (per risparmiare memoria) ma possono essere portati a 16 selezionando nel menu File l'opzione ColoriSchermo. Anche se KW sta lavorando con 4 colori attraverso l'item ColoriImmagine del menù Extra è sempre possibile osservare l'immagine nei suoi colori originali.

Durante la stampa, nel caso si usi una stampante a colori, il disegno verrà stampato con i suoi colori originali indipendentemente da come appare sullo schermo.

Le immagini, una volta inserite, possono venir facilmente manipolate: è possibile infatti, oltre a spostare o duplicare le stesse in qualsiasi punto del documento, muoverle orizzontalmente, modificarne le dimensioni o ritagliarle sui quattro lati e questo è veramente notevole per un programma di wordprocessor.

CMB__MPS1000.PRN
Epson.PRN
Epson24.PRN
Epson__JX-80.PRN
ImagewriterII.PRN
Okimate__20.PRN

Marca	Stampante	Driver
Apple	Imagewriter II	ImagewriterII
Apple	Imagewriter	ImagewriterII
Brother	1509	Epson (1)
CIE	CI-3500	CBM__MPS1000
CIE	Lips 10	Epson (1)
Citizen	serie MSP	Epson
Commodore	MPS-1000	CBM MPS1000
Epson	Epson	Epson
Epson	serie FX	Epson
Epson	JX-80	Epson__JX-80
Epson	serie LX	Epson
Epson	serie MX	Epson
Epson	serie RX	Epson
Juki	5510	Epson__JX-80
NEC	CP2 e CP3	Epson__JX-80
Okidata	Okimate 20	Okimate 20
Okidata	192 e 193	CBM__MPS1000 (2)
Okidata	292 e 293	CBM__MPS1000 (2)
Panasonic	1080	Epson
Panasonic	1091	Epson
Panasonic	1092	Epson
Panasonic	1592	Epson
Star	tutte	Epson (1) o CBM__MPS1000

(1) Stampante in modo Epson.
(2) Stampante in modo IBM.

Tabella stampanti

Funzioni avanzate

Come abbiamo già accennato è possibile selezionare attraverso il menu Carattere tre tipi di font: Roman 8, Roman 12 e Roman 14 e dei caratteri più piccoli da usare come apici o pedici ed inoltre tutta una serie di simboli particolari tra cui le linee per la costruzione di tabelle. I vari font inoltre possono comparire nei vari stili di normale, corsivo, grassetto o sottolineato.

Un particolare molto importante, nel caso di un documento scritto in vari tipi di carattere, è che se viene inserito altro testo in quello esistente, la parte inserita apparirà nello stesso tipo di font e stile. Il carattere attivato sarà sempre quello direttamente a destra del cursore.

Altra caratteristica di rilievo di KW è la possibilità di creare intestazioni o piè di pagina. Selezionando le stesse da menu, appare una nuova finestra con un righello separato che può contenere fino a 15 righe di testo. Nella nuova finestra è possibile inserire la data e l'ora (prelevata dalle Preferences) ed anche la numerazione automatica delle pagine con diversi stili (romano, arabo eccetera)

Dalle finestre di cui sopra è possibile aprire i menu che ricalcano, con piccole varianti, i menu della finestra principale di KW.

Nelle intestazioni o nei piè di pagina possono venir naturalmente usati tipi di caratteri o formattazioni diverse da quelle del documento principale. Il testo, all'interno di esse, e che appare nel documento principale, può venir modificato in seguito solamente aprendo le corrispondenti finestre.

Con un'apposita opzione del menù principale si può operare in modo che il testo dell'intestazione o del piè di pagina non compaia sulla prima pagina del documento.

Proseguendo nell'esaminare KW vediamo che esso dispone come molti altri wordprocessor della funzione di ricerca e sostituzione del testo. In questo programma è consentito anche il rispetto delle lettere maiuscole e, tramite un'opzione, la ricerca di una stringa che compare nel documento solo come parola singola e non come parte di un'altra parola.

Mail-merge

Il mail-merge è un comando utile per creare molte versioni di un unico documento che differiscono tra loro solo per qualche particolare come indirizzo, nominativo, eccetera. Abbiamo già accennato in precedenza al caso della lettera da inviare a più persone.

In KW le parole del testo che devono venir sostituite vengono evidenziate con due coppie di parentesi triangolari poste ai lati delle parole stesse. Ritornando all'esempio della lettera sarà sufficiente introdurre la parola <<indirizzo>> nella posizione voluta. Dopo aver salvato il testo della lettera sul dischetto, si dovrà creare un file (lista di stampa) con i nomi e gli indirizzi dei destinatari. In fase di stampa, dopo aver selezionato la funzione Mail-merge dal menu File, il programma sostituirà automaticamente <<indirizzo>> con ciascuno degli effettivi indirizzi prelevati dalla lista di stampa. La creazione della lista di stampa deve seguire alcune precise regole, cioè i dati devono essere inseriti con dei caratteri di controllo particolari, ma il tutto risulta molto semplice ed è descritto dettagliatamente nel manuale d'uso

Andiamo in stampa

KW stampa il documento così come appare sullo schermo e, se possedete una stampante a colori, le immagini verranno stampate a colori.

Le modalità di stampa selezionabili da menu sono:

Bozza: stampa veloce in bassa risoluzione; non sempre corrisponde esattamente a quello che si vede sullo schermo;

Lettera: utilizza i caratteri NLQ della stampante oppure i caratteri Superfont per tutti quei simboli o caratteri particolari che la stampante non riesce a riconoscere;

Superfonts: è una funzione speciale che permette di stampare il documento in alta risoluzione con un'elevata qualità di stampa anche con stampanti piuttosto economiche, naturalmente a scapito della velocità che risulta essere abbastanza ridotta. Le stampanti devono comunque essere a matrice di punti.

Quando si stampa con Superfonts, il programma controlla il driver di stampa installato nelle Preferences e quindi cerca il proprio speciale driver di stampa sul disco di KindWords. Questi driver hanno lo stesso nome di quelli delle Preferences ma con l'estensione "PRN" e sono contenuti nella directory "SuperDrivers". Esaminate la tabella per conoscere quali stampanti sono compatibili con KW. A detta del manuale comunque è previsto che il programma possa usare in futuro molte altre stampanti.

Conclusioni

Che dire di KindWords, in definitiva si presenta come un programma molto completo che soddisferà senz'altro le esigenze degli utenti più svariati i quali lo potranno usare sia per la semplice lettera che per la stesura di libri o documenti professionali. La possibilità di inserire immagini apre ulteriori campi di utilizzo anche se non siamo certo ai livelli di un DTP.

La semplicità d'uso è senz'altro il pezzo forte di questo programma che in pratica può venir usato immediatamente senza neanche aprire il manuale ma dando una rapida occhiata ai vari menu.

Tra i difetti possiamo annoverare solo la mancanza di ulteriori font, di non prelevare l'ora e la data direttamente dall'orologio interno e di non poter aprire più documenti contemporaneamente.

La versione inglese del programma viene fornita con il controllo ortografico, cosa che non è stata fatta per la versione italiana. Non possiamo escludere comunque, se il programma incontrerà i favori del pubblico, di vederne la comparsa in un futuro non troppo lontano.

LISTINO LIBRI JACKSON

CODICE	TITOLO	PREZZO	
	INFORMATICA: CONCETTI GENERALI		
511 A	COME PROGRAMMARE	15.000	
503 A	PROGRAMMAZIONE STRUTTURATA, CORSO DI AUTOISTRUZIONE	15.000	
101 H	TERMINI DELL'INFORMATICA E DELLE DISCIPLINE CONNESSE	50.000	
539 A	LOGICA E DIAGRAMMI A BLOCCHI: TECNICHE DI PROGRAMMAZIONE	40.000	
526 P	DATA BASE: CONCETTI E DISEGNO	22.500	
GYS190	TRADUTTORI DI LINGUAGGI	26.000	
G 240	PAROLE BASE DELL'INFORMATICA	8.000	
GYS245	CONCETTI DI INFORMATICA	43.000	
GYS248	DATA PROCESSING	45.000	
GY 264	DATA FILE	50.000	
GYS266	ARCHITETTURE DI SISTEMA	32.000	
GY 354	SISTEMI INTELLIGENTI	28.000	
CZ 419	ANALISI E PROGRAMMAZIONE	11.000	
158 EC	INFORMATICA DI BASE I CONCETTI FONDAMENTALI HARDWARE E SOFTWARE	55.000	
526 A	VOI E L'INFORMATICA	15.000	
100 H	DIZIONARIO DI INFORMATICA	59.000	
GY 551	I LINGUAGGI DELLA 4ª GENERAZIONE	65.000	
GYS552	PRIMA DEL LINGUAGGIO LA PROGRAMMAZIONE	35.000	
GYS 559	C.S.P. - PROCESSI SEQUENZIALI	49.000	
GYS 546	ALGORITMI FONDAMENTALI	54.000	
GY 618	SISTEMI ESPERTI	28.000	
047 T	MICROPROCESSORI	14.500	
048 T	DATA BASE	14.500	
049 T	FILE	14.500	
CI 686	CAPIRE IL PERSONAL COMPUTER	35.000	
G 540	MODELLI MATEMATICI E SIMULAZIONE	56.000	
GE 688	ENCICLOPEDIA MONOGRAFICA DI ELETTR. E INF. VOLUME I	58.000	
GE 689	ENCICLOPEDIA MONOGRAFICA DI ELETTR. E INF. VOLUME II	58.000	
GY 629	SOFTWARE DI BASE - Strumenti di sviluppo	52.000	
	INFORMATICA: SISTEMI OPERATIVI		
G 223	UNIX LA GRANDE GUIDA	70.000	
GY 272	SISTEMI OPERATIVI PER MICROCOMPUTER	25.000	
GY 273	MS-DOS LA GRANDE GUIDA	45.000	
510 P	CP/M CON MP/M	29.000	
CZ 538	MS DOS 2 E 3	49.000	
G 543	XENIX	45.000	
R 588	LAVORARE CON XENIX	70.000	
GYS271	SISTEMI OPERATIVI	55.000	
R 615	I COMANDI DI XENIX MAIL	12.500	
092 D	SOFTWARE DI BASE E SISTEMI OPERATIVI	7.000	
093 D	CP/M IL "SOFTWARE BUS"	7.000	
094 D	MS-DOS E PC-DOS LO STANDARD IBM	7.000	
009 H	UNIX	8.500	
011 H	CP/M	8.500	
044 T	MS DOS	14.500	
045 T	PC DOS	14.500	
R 628	MICROSOFT OS/2	50.000	
046 T	UNIX	14.500	
MS 02 E	COFANETTO "MS-DOS" 5"¼- Corso autoistruzione	156.000	F
R 600	MS DOS ADVANCED - Il Manuale del Programmatore	55.000	
GY 663	UNIX PROGRAMMAZIONE AVANZATA	55.000	
BY 724	GUIDA AI SISTEMI OPERATIVI	29.000	
BY 744	UNIX: CONCETTI, STRUTTURE, UTILIZZO	43.000	
R 761	MS-DOS REFERENCE GUIDE	14.500	
	INFORMATICA: LINGUAGGI		
501 A	IMPARIAMO IL PASCAL	16.000	
502 A	INTRODUZIONE AL BASIC	25.000	
500 P	PASCAL MANUALE E STANDARD DEL LINGUAGGIO	16.000	
329 A	PROGRAMMARE IN ASSEMBLER	14.000	
513 A	PROGRAMMARE IN BASIC	8.000	
514 A	PROGRAMMARE IN PASCAL	19.000	
516 A	INTRODUZIONE AL PASCAL	39.000	
517 P	DAL FORTRAN IV AL FORTRAN 77 (II ED.)	32.000	
521 A	50 ESERCIZI IN BASIC	17.000	
525 A	BASIC PER TUTTI	23.000	

CODICE	TITOLO	PREZZO	
534 A	MANUALE DEL BASIC	45.000	
509 A	LOGO: POTENZA E SEMPLICITÀ	20.500	
507 B	TUO PRIMO PROGRAMMA IN BASIC (II)	19.500	
533 A	BASIC DALLA A ALLA Z	19.000	
540 A	LINGUAGGIO ADA	19.500	
541 P	LINGUAGGIO C	25.000	
542 P	COBOL STRUTTURATO: CORSO DI AUTOISTRUZIONE	50.000	
508 P	PROGRAMMARE IN C	39.000	
G 233	COBOL PER MICROCOMPUTER	35.000	
GYS246	ESERCIZI DI FORTRAN	20.000	
GYS247	ESERCIZI IN PASCAL: ANALISI DEI PROBLEMI	29.000	
GYS254	PROGRAMMAZIONE IN LINGUAGGIO ADA	42.000	
GY 270	APL PER IL P.C. IBM	25.000	
GYS274	DAL PASCAL AL MODULA 2	26.000	
GYS311	LINGUAGGIO C IL LIBRO DELLE SOLUZIONI	24.000	
GYS328	APPLICAZIONI IN PASCAL	32.000	
GY 535	TURBO PASCAL	29.000	
G 544	"C" LIBRARY	49.000	
GYS550	PROLOG - LINGUAGGIO E APPLICAZIONE	32.000	
R 589	TURBOPASCAL - LIBRERIA DI PROGRAMMI	45.000	
042 T	LINGUAGGIO C	12.500	
108 D	FORTH ANATOMIA DI UN LINGUAGGIO	7.000	
107 D	FORTRAN E COBOL LINGUAGGI SEMPRE VERDI	7.000	
086 D	ED È SUBITO BASIC VOL. 1	7.000	
087 D	ED È SUBITO BASIC VOL. 2	7.000	
034 T	PROLOG	14.000	
035 T	LISP	12.500	
001 H	COBOL	8.500	
006 H	PASCAL	8.500	
007 H	BASIC	8.500	
010 H	FORTRAN 77	8.500	
020 H	LOGO	8.500	
022 H	FORTH	8.500	
R 612	TURBO PROLOG	50.000	
GY 626	IL MANUALE DEL PASCAL	42.000	
GY 616	DEBUGGING C	55.000	
GY 687	DALLA PROGRAMMAZIONE STRUTTURATA AL PASCAL	42.000	
GY 634	FONDAMENTI DI COMMON LISP	40.000	
	INFORMATICA: LAVORO E SOCIETÀ		
519 P	COMPUTER GRAFICA	29.000	
800 P	ODISSEA INFORMATICA	50.000	
407 H	APPLICAZIONI DEL COMPUTER NELL'UFFICIO MODERNO	23.000	
802 H	INFORMATICA MUSICALE	27.000	
802 P	COMPUTERGRAPHIA	40.000	
806 P	COMPUTER PER L'INGEGNERIA EDILE	22.000	
807 P	COMPUTER PER IL MEDICO	19.000	
CI 231	COMPUTER IMAGE	40.000	
CI 241	ODISSEA INFORMATICA STRATEGIE CULTURALI PER UNA SOCIETÀ INF.	32.000	
G 400	COMPUTER GRAPHICS E ARCHITETTURA	27.000	
PV 409	COMPUTER GRAPHICS E MEDICINA	18.000	
GY 487	MEDICO & COMPUTER	45.000	
GY 548	INFORMATICA MEDICA	65.000	
PA 685	OFFICE AUTOMATION	28.000	
RA 596	DESKTOP PUBLISHING	35.000	
050 T	WORD	14.500	
	INFORMATICA: SOFTWARE PACCHETTI APPLICATIVI		
570 P	CONTABILITÀ COL PERSONAL COMPUTER	27.000	
525 P	WORDSTAR	24.000	
546 P	MANUALE DEL DBASE II	24.000	
578 P	PC NELL'ORG. DELLE PICCOLE AZIENDE: APPL. DEL MULTIPLAN	29.000	F
PP 219	LOTUS 1-2-3: GUIDA ITALIANA ALL'USO	21.000	
G 234	RIORDINO E GESTIONE DEGLI ARCHIVI APPLICAZIONI CON PFS-FILE	30.000	
PP 255	DBASE III GUIDA ITALIANA ALL'USO	45.000	
PA 282	MODELLI DECISIONALI PER IL MANAGER	50.000	
PA 288	PIANIFICAZIONE AZIENDALE PLANNING, MARKETING STRAT., BUDGETING	35.000	F
PP 310	LA GRANDE GUIDA LOTUS A SYMPHONY	70.000	
PP 326	MULTIPLAN CORSO D'ISTRUZIONE	40.000	F

CODICE	TITOLO	PREZZO	
PP 344	FRAMEWORK II - GUIDA ITALIANA ALL'USO	27.000	
PP 351	WORD PROCESSING	27.000	
PP 467	IMPARA 1-2-3 CON LA GRANDE GUIDA LOTUS	45.000	
PP 468	CHART - CORSO ISTRUZIONE	45.000	F
PP 473	IL NUOVO 1-2-3 GUIDA ALL'USO DELLA VERSIONE ITALIANA 2 LOTUS 1-2-3	29.000	
PA 474	BILANCIO, BUDGET, CASH FLOW	40.000	F
PP 475	DBASE III - CORSO DI PROGRAMMAZIONE	23.000	
PA 476	PREVISIONE, PIANIFICAZIONE, SIMULAZIONE CON LOTUS 1-2-3	60.000	F
PV 477	GUIDA ALLA BUSINESS GRAPHIC	20.000	
PP 480	AUTOCAD	40.000	
PP 481	RBASE 5000 - GUIDA ITALIANA ALL'USO	20.000	
PP 537	IL MANUALE DI WINDOWS	60.000	
PP 539	DBASE III - TECNICHE AVANZATE DI PROGRAMMAZIONE	42.000	
PP 545	APPLICAZIONI DI DBASE III	50.000	F
PA 566	MODELLI DECISIONALI CON LOTUS 1-2-3	40.000	F
PP 577	MANUALE DBASE III PLUS	49.000	
039 T	WORDSTAR	12.500	
040 T	LOTUS 1-2-3	12.500	
043 T	WINDOWS	12.500	
PP 621	I COMANDI DI DBASE III PLUS	12.500	
095 D	GUIDA AI PACKAGE APPLICATIVI MERCEOLOGIA DEL SOFTWARE	7.000	
096 D	VISICALC GUIDA RAPIDA ALL'UTILIZZO	7.000	
098 D	WORD PROCESSING	7.000	
103 D	LOTUS 1-2-3 E SIMPHONY IL FASCINO DELL'INTEGRAZIONE	7.000	
104 D	DBASE II E III I PRINCIPI DI DATABASE	7.000	
106 D	MULTIPLAN SPREADSHEET MULTISTRATO	7.000	
110 D	PACKAGE A CONFRONTO PROVE DEI SOFTWARE PIÙ DIFFUSI	7.000	
031 T	FRAMEWORK E FRAMEWORK II	12.500	
033 T	MULTIPLAN 2.02	12.500	
036 T	SYMPHONY	12.500	
038 T	REFLEX	12.500	
027 H	EASY SCRIPT	8.500	
033 H	PAGE MAKER	8.500	
034 H	PROJECT	8.500	
035 H	RBASE	8.500	
PP 611	GUIDA ALL'USO PROFESSIONALE REFLEX	55.000	
PP 636	MANUALE DI WORD	70.000	
PP 594	GUIDA ALL'USO PROFESSIONALE DI LOTUS 1-2-3	50.000	
PP 593	VENTURA - Il grande manuale	55.000	
R 671	LINGUAGGIO C - Reference guide	12.500	
051 T	I COMANDI DI LOTUS 1-2-3 - Reference guide	12.500	
PP 581	PROGRAMMARE IN FRED	40.000	
PP 631	dBASE III PLUS - Guida uso professionale	65.000	
PP 694	PROGRAMMARE IN WINDOWS	70.000	
PA 592	GESTIONE DELLA PRODUZIONE	40.000	F
PP 727	VENTURA - REFERENCE GUIDE	14.500	
PP 700	MATEMATICA CON LOTUS 1-2-3	35.000	F
R 574	MANUALE DELLE STAMPANTI LASER	25.000	
PP 641	AUTOCAD - Il grande manuale	55.000	
PP 728	VENTURA - Fogli stile	42.000	F
PP 741	WORD 3 e 4	59.000	F
PP 642	AUTOCAD Programmazione avanzata	65.000	F
BY 707	ORACLE	75.000	
PA 771	MODELLI PER LOTUS 1-2-3	28.000	F
	PERSONAL COMPUTER		
550 D	PROGRAMMI PRATICI IN BASIC	15.000	
515 H	BASIC E LA GESTIONE DEI FILE VOL. I: METODI PRATICI	15.000	
551 D	75 PROGRAMMI IN BASIC PER IL VOSTRO COMPUTER	12.000	
552 D	PROGRAMMI DI MATEMATICA E STATISTICA IN BASIC	20.000	
554 P	PROGRAMMI SCIENTIFICI IN PASCAL	29.000	
516 H	BASIC E LA GESTIONE DEI FILE - VOL. 2	17.000	
CH 182	COMPUTER HARDWARE REALIZZ. PRATICHE PER GLI HC PIÙ DIFFUSI	18.000	
CI 187	COMPUTER L'HOBBY E IL LAVORO	12.000	
G 235	GRAFICA PER PERSONAL COMPUTER	39.000	
GE 263	METODI DI INTERFACC. PERIFERICHE	43.000	
GE 402	CORSO DI AUTOISTRUZIONE PER MICROCOMPUTER VOL. 1 + VOL. 2	35.000	
PA 406	COME GESTIRE LA PICCOLA AZIENDA CON IL P.C.	22.000	
PP 408	BUSINESS IN BASIC	23.000	
CI 412	IL COMPUTER È UNA COSA SEMPLICE	15.000	

CODICE	TITOLO	PREZZO
CC 415	CONTROLLO DEI DISPOSITIVI DOMESTICI CON IL P.C.	23.000
159 GC	PERSONAL COMPUTER DAL SOFTWARE DI BASE ALLE APPLICAZIONI D'UFFICIO	55.000
R 587	HARD DISK - LA GRANDE GUIDA	75.000
084 D	INTRODUZIONE AI PERSONAL COMPUTER VIVERE COL PC	7.000
099 D	SCRIVERE UN'AVVENTURA, 1000 AVVENTURE COL PROPRIO PC	7.000
100 D	GRAFICA E BASIC LE BASI DELLA COMPUTERGRAFICA	7.000
085 D	HARDWARE DI UN PERSONAL COMPUTER DENTRO E FUORI LA SCATOLA	7.000
101 D	GESTIONE DEI FILE IN BASIC E PASCAL VOL. 1	7.000
102 D	GESTIONE DEI FILE IN BASIC E PASCAL VOL. 2	7.000
113 D	DISEGNARE COL PERSONAL COMPUTER	7.000
105 D	PERSONAL E HOME COMPUTER A CONFRONTO	7.000
112 D	SUONO E MUSICA COL PERSONAL COMPUTER	7.000
109 D	COSTRUIRSI UN PERSONAL DATABASE	7.000
097 D	GUIDA ALL'ACQUISTO DI UN PERSONAL COMPUTER	7.000
088 D	TO DO OR NOT TO DO COME AVER CURA DEL PROPRIO PC	7.000
089 D	SOFTWARE STRUTTURATO CON ELEMENTI DI PASCAL	7.000
090 D	DIZIONARIO DI INFORMATICA	7.000
091 D	BASI DELLA PROGRAMMAZIONE STENDERE UN PROG. COME SI DEVE	7.000
004 H	PROGRAMMAZIONE	8.500
015 H	PROGRAMMI DI STATISTICA	8.500
	PERSONAL COMPUTER: COMMODORE	
347 D	VOI E IL VOSTRO COMMODORE 64	24.000
348 D	COMMODORE 64 - IL BASIC	28.000
400 D	FACILE GUIDA AL COMMODORE 64	13.500
400 B	COMMODORE 64 - FILE	19.000
409 B	COMMODORE 64 - LA GRAFICA E IL SUONO	34.000 C
570 D	MATEMATICA E COMMODORE 64	26.500 C
350 D	LIBRO DEI GIOCHI DEL COMMODORE 64	24.000 C
575 D	TECNICHE DI PROGRAMMAZIONE SUL COMMODORE 64	16.500
572 D	LINGUAGGIO MACCHINA DEL COMMODORE 64	35.000 F
576 D	SISTEMA TOTOMAC: LA NUOVA FRONTIERA DEL TOTOCALCIO	29.000 C
548 B	64 PERSONAL COMPUTER E C64	45.000
SDP222	STATISTICA AD UNA DIMENSIONE CON IL C64	24.000
CC 260	AVVENTURE (COMMODORE 64)	20.000 C
CC 320	AMIGA HANDBOOK	35.000
CC 322	COMMODORE 128 OLTRE IL MANUALE	29.000
CC 323	PROGRAMMI PER COMMODORE 128	29.000 C
CZ 541	128 E 64 - LE PERIFERICHE	32.000
CC 564	MANUALE RIPARAZIONE C64	55.000
CZ 532	MANUALE DI AMIGA	39.000
002 H	COMMODORE 64	8.500
CC 658	GRAFICA E SUONO PER C64 - 64PC - C128	35.000 F
CC 657	MANUALE DEL COMMODORE C64 - C64PC - C128	35.000 F
CC 627	AMIGA 500	55.000
CC 750	C.128 LA GRANDE GUIDA	50.000
CC 749	C.64 LA GRANDE GUIDA	50.000
	PETER NORTON	
R 734	MANUALE DEL DOS	55.000
R 736	INSIDE PC IBM	63.000 I
R 733	HARD DISK COMPANION	60.000
R 735	LINGUAGGIO ASSEMBLY PER PC IBM	72.000 2F
	PERSONAL COMPUTER: IBM	
564 D	PROGRAMMI UTILI PER IBM PC	19.000
G 217	GRAFICA PER IL PERSONAL COMPUTER IBM	39.000
GY 319	PC IBM MANUALE DEL LINGUAGGIO MACCHINA	46.000
GY 335	MAPPING PC IBM GESTIONE DELLA MEMORIA	42.000
PP 407	MANUALE BASE DEL PC IBM	22.000
041 T	PC IBM	12.500
R 609	SOLUZIONI AVANZATE PER IL PROGRAMMATORE	60.000
CZ 751	AVVENTURE PER MS-DOS	36.000 F
RA 484	GUIDA ALLE RETI DI PC IBM	46.000

CODICE	TITOLO	PREZZO
	PERSONAL COMPUTER: OLIVETTI	
401 P	PRIMO LIBRO PER M24: MS DOS E GW BASIC	28.000
401 B	OLIVETTI M10: GUIDA ALL'USO	18.000
CL 216	BASIC IN 30 ORE PER M24 ED M20	32.000
CZ 483	MANUALE OLIVETTI M19	42.000
CZ 536	MANUALE PC 128 OLIVETTI PRODEST	29.000
CZ 582	PROGR. PER PC 128 OLIVETTI PRODEST (CASS.)	27.000
	PERSONAL COMPUTER: MSX	
CZ 181	30 PROGRAMMI PER MSX	20.000 C
417 D	MSX: IL BASIC	23.000
CC 261	AVVENTURE (MSX)	20.000 C
CC 289	SUPER PROGRAMMI PER MSX	35.000 C
CC 336	MSX LA GRAFICA	25.000
111 D	STANDARD MSX	7.000
	PERSONAL COMPUTER: APPLE	
331 P	APPLE II GUIDA ALL'USO	31.000
416 P	MACINTOSH NEGLI AFFARI: MULTIPLAN E CHART	16.500
424 P	UN MAC PER AMICO: USO, APPLICAZIONI E PROGRAMMI PER MACINTOSH	12.000
PP 224	MACINTOSH ARTISTA: MACPAINT E MACDRAW	16.000
CCP277	APPLE IIC GUIDA ALL'USO	45.000
CC 312	PROGRAMMI PER APPLE IIC	13.000
CC 417	PROGRAMMI COMM. E FINANZIARI CON APPLE	22.000
340 H	APPLE MEMO	15.000
CC 576	IL MANUALE DELL'APPLE II GS	28.000
003 H	APPLE IIE IIC	8.500
CC 665	MICROSOFT BASIC PER APPLE MACINTOSH	32.000 F
	PERSONAL COMPUTER: ATARI - AMSTRAD - SHARP	
540 H	BASIC ATARI	18.000
CC 330	PROGRAMMI PER AMSTRAD CPC 464 CPC 664 - CPC 6128	29.000 C
CC 331	PROGRAMMI PER ATARI 130XE	19.000 C
CC 471	MANUALE ATARI 520 ST E 1040 ST	28.000
CC 486	WORD PROCESSING CON AMSTRAD PCW 8256/8512	35.000
032 T	AMSTRAD PCW 8256 e PCW 8512	14.000
028 H	AMSTRAD 464 E 664	8.500
	COMMUNICATION E TELEMATICA	
309 A	PRINCIPI E TECNICHE DI ELABORAZIONE DATI	20.000
518 D	TELEMATICA	28.000
528 P	TRASMISSIONE DATI	27.000
617 P	RETI DATI: CARATTERISTICHE, PROGETTO E SERVIZI TELEMATICI	40.000
GYS314	ELABORAZIONE DIGITALE DEI SEGNALI: TEORIA E PRATICA	25.000
PA 327	BANCHE DATI RICERCA ONLINE	26.000
158 LC	COMUNICAZIONI DALLE ONDE ELETTROMAGNETICHE ALLA TELEMATICA	55.000
CC 472	MODEM E PC USO E APPLICAZIONI	25.000
GTS478	RETI LOCALI	44.000
GTS479	IL MODEM - TEORIA, FUNZIONAMENTO	28.000
R 542	TRASMISSIONE DATI PER PC	31.000
GT 555	LA TELEMATICA NELL'UFFICIO	35.000
R 601	COLLEGAMENTO TRA MICRO E MAINFRAME	39.000
BT 655	MANUALE DI TV E VIDEO COMMUNICATION	45.000
	ELETTRONICA DI BASE E TECNOLOGIA	
201 A	CORSO DI ELETTRONICA FONDAMENTALE CON ESPERIMENTI	35.000
204 A	ELETTRONICA INTEGRATA DIGITALE	50.000
205 A	MANUALE PRATICO DI PROGETTAZIONE ELETTRONICA	35.000
200 A	SISTEMI DIGITALI: MANUTENZIONE, RICERCA ED ELIMINAZIONE GUASTI	28.500
GES262	TECNOLOGIE VLSI	70.000
GES390	ELETTRONICA INTEGRATA DIGITALE IL LIBRO DELLE SOLUZIONI	17.000
CE 411	LA FISICA DEI SEMICONDUTTORI	10.000
158 PC	ELETTRONICA DI BASE I FONDAMENTI DELL'ELETTRONICA ANALOGICA	55.000
158 CC	ELETTRONICA DIGITALE VOL. 1 DALLE PORTE LOGICHE AI CIRCUITI INTEGRATI	55.000
158 DC	ELETTRONICA DIGITALE VOL. 2 DAI BUS AI GATE ARRAY	55.000
158 GC	ELETTROTECNICA ELETTROSTATICA ELETTROMAGNETISMO RETI ELETTR.	55.000

CODICE	TITOLO	PREZZO
	ELETTRONICA APPLICATA	
601 B	TIMER 555	10.000
203 A	INTRODUZIONE AI CIRCUITI INTEGRATI DIGITALI	10.000
612 P	MANUALE DEGLI SCR VOL. 1	28.000
613 P	MANUALE DI OPTOELETTRONICA	15.000
614 A	FIBRE OTTICHE	15.000
GE 403	JFET MOS E DATA BOOK	20.000
GE 404	TRANSISTOR DATA BOOK	32.000
GE 405	METODI DI PROTEZIONE CONTRO LE SOVRATENSIONI	17.000
CE 413	IL MANUALE DEGLI SCR E TRIAC	15.000
CE 421	MANUALE DEI FILTRI ATTIVI	29.000
CE 423	MANUALE DEI PLL PROGETTAZIONE DEI CIRCUITI	29.000
CE 425	MANUALE DEGLI AMPLIFICATORI OPERAZIONALI	29.000
CE 429	250 PROGETTI CON GLI AMPLIFICATORI DI NORTON	39.000
CE 431	MANUALE DEI CMOS	25.000
CE 485	IL COLLAUDO DELLE SCHEDE	18.000
BE 557	I TRASDUTTORI	43.000
BT 585	FIBRE OTTICHE	29.000
BE 578	MANUALE DI ELETTRONICA	29.000
BE 558	IL MANUALE DEL TECNICO ELETTRONICO	51.000
BE 610	GUIDA ALLA STRUMENTAZIONE ELETTRONICA	34.000
BE 619	MULTIMETRI DIGITALI	42.000
BE 639	ENCICLOPEDIA DEI CIRCUITI INTEGRATI	60.000
BE 654	MANUALE DI ELETTRONICA DEL COMPUTER	20.000
701 P	MANUALE PRATICO DEL RIPARATORE RADIO TV	29.000
705 P	IMPIEGO PRATICO DELL'OSCILLOSCOPIO	17.500
615 P	PROGETTAZIONE DI SISTEMI DI ALTOPARLANTI	21.000
CE 427	L'ELETTRONICA A STATO SOLIDO	25.000
BE 718	77 SCHEDE PER IL RIPARATORE TV	40.000
BE 723	MISURE DEI CIRCUITI ELETTRONICI	26.000
BE 721	MANUALE PRATICO DI ELETTRONICA DIGITALE	26.000
BE 684	IL MANUALE DEI CMOS	35.000
BE 731	IL MANUALE DEGLI AMPLIFICATORI OPERAZIONALI	39.000
	ELETTRONICA: MICROPROCESSORI	
310 P	NANOBOOK Z80 VOL. 1	20.000
007 A	BUGBOOK VII	17.000
314 P	TECNICHE DI INTERFACCIAMENTO DEI MICROPROCESSORI	31.000
312 P	NANOBOOK Z80 VOL. III	25.000
320 P	MICROPROCESSORI DAI CHIPS AI SISTEMI	29.000
324 P	PROGRAMMAZIONE DELLO Z80 E PROGETTAZIONE LOGICA	21.500
326 P	Z80 PROGRAMMAZIONE IN LINGUAGGIO ASSEMBLY	50.000
328 D	PROGRAMMAZIONE DELLO Z80	40.000
504 B	APPLICAZIONI DEL 6502	17.000
503 B	PROGRAMMAZIONE DEL 6502	35.000
505 B	GIOCHI CON IL 6502	19.500
G 220	8086-8088 PROGRAMMAZIONE	40.000
GY 265	ASSEMBLER PER IL 68000	70.000
CE 410	IMPIEGO DELLO Z80	23.000
158 HC	MICROPROCESSORI ARCHIT. PROGR. E INTERFAC. DEI MP DA 4 A 32 BIT	55.000
013 H	ASSEMBLER 6502	8.500
016 H	ASSEMBLER Z80	8.500
021 H	ASSEMBLER 68000	8.500
025 H	ASSEMBLER 8086-8088	8.500
029 H	ASSEMBLER 80286	8.500
GE 567	80286 ARCHITETTURA E PROGRAMMAZIONE	58.000
GY 603	80386 ARCHITETTURA E PROGRAMMAZIONE	37.000
	AUTOMAZIONE	
208 A	CONTROLLORI PROGRAMMABILI	24.000
616 P	CONTROLLO AUTOMATICO DEI SISTEMI	29.500
GES251	STRUTTURA E FUNZIONAMENTO DEI CONTROLLI NUMERICI	29.000
GES252	CONTROLLI NUMERICI: PROGRAMMAZIONE E APPLICAZIONI	28.000
G 399	30 APPLICAZIONI DI CAD	29.000
G 401	CAD/CAM & ROBOTICA	28.000
CI 414	DAL CHIP ALLA ROBOTICA	15.000
GE 547	LA PROGETTAZIONE AUTOMATICA	32.000
GE 564	ROBOTICA - Fondamenti e applicazioni	38.000

CODICE	TITOLO	PREZZO	
	DIZIONARI ENCICLOPEDICI		
DS 498	FISICA	14.000	
DS 499	MATEMATICA	14.000	
DS 522	GEOLOGIA	14.000	
DS 524	ELETTRONICA	14.000	
DS 525	ASTRONOMIA	14.000	
DS 526	CHIMICA	14.000	
DS 527	RAGIONERIA GENERALE	14.000	
DS 528	RAGIONERIA APPLICATA	14.000	
DS 529	BIOLOGIA	14.000	
DS 530	MECCANICA	14.000	
DS 531	INFORMATICA	14.000	
	ARGOMENTI VARI		
704 D	MANUALE PRATICO DI REGISTRAZIONE	10.000	
706 A	COMUNICAZIONI RADIO IN MARE	18.000	
800 H	FENDER, STORIA DI UN MITO	28.000	
CZ 748	TOTOCALCIO, ENALOTTO, TOTIP	35.000	F
	SOFTWARE E MANAGEMENT TOOLS		
CZ 469	GRAFIX - DISEGNARE CON IL PC	50.000	F
TP 606	CORSO AUTOISTRUZIONE LOTUS 1-2-3 (VERS. ITALIANA) F - MS DOS	90.000	F
TY 605	CORSO AUTOISTRUZIONE SUL SISTEMA MS DOS - FLOPPY	50.000	F
TY 640	TURBO PASCAL - LIBRERIA DI PROGRAMMI	40.000	F
TP 643	CORSO AUTOISTRUZIONE LOTUS 1-2-3 (INGLESE)	90.000	F
TP 608	BUDGET STRATEGICO (LOTUS 1-2-3)	100.000	F
TP 614	GESTIONE DELLE COMMESSE DI PRODUZIONE	100.000	F
TP 623	CONTROLLO DELLE VENDITE (CON MULTIPLAN)	100.000	F
TP 625	GESTIONE DEL PERSONALE (LOTUS 1-2-3)	100.000	F
TP 677	GESTIONE DELLE COMMESSE CON MULTIPLAN 2.0	100.000	F
TP 673	PREVENTIVO E CONSUNTIVO DEI COSTI - CON LOTUS 1-2-3 VERS. 2 E MULTIPLAN 2.0	100.000	F
TP 660	1-2-3 LIBRERIA DI MACRO	60.000	F
TY 691	SUPER SCREEN - UTILITY PER I PROGRAMMATORI	50.000	F
TY 690	PC DOCTOR UTILITY - RECOVERING DEI FILE	60.000	F
TP 644	STATISTICA A UNA E DUE DIMENSIONI	100.000	F
TP 681	ANALISI ABC CON LOTUS 1-2-3	100.000	F
TP 669	GESTIONE DELLE COMMESSE CON dBASE III PLUS	100.000	F
	MARKETING & MANAGEMENT		
M 648	PROBLEMI DI MARKETING	45.000	
M 649	DISTINTA BASE	23.000	
M 650	TECNICHE DI ANALISI FINANZIARIA	52.000	
M 647	RICERCHE DI MERCATO	72.000	
	NOVITÀ NOVEMBRE '88		
BE 737	IL MANUALE DEI FILTRI ATTIVI	34.000	
R 730	TURBO C	62.000	F
GY 662	UNIX - Architettura di sistema	65.000	
PP 672	MARKETING CON LOTUS 1 2 3	41.000	F
GY 703	LINGUAGGI DI PROGRAMMAZIONE	69.000	
BE 713	80286 HARDWARE	65.000	
M 679	DIRECT MARKETING	35.000	
M 706	TECNICHE DI MARKETING	43.000	
M 726	DIRECT MAILING	35.000	

F = libro con floppy
C = libro con cassette

Per le vostre ordinazioni per corrispondenza utilizzate l'apposita cedola inserita in questa rivista.

YD008/D

JACKSON PRESENTA

ATAX

IL FANTASTICO VIDEOGAME INEDITO IN ANTEPRIMA MONDIALE REALIZZATO APPOSITAMENTE DALLA INTERTEINMENT INTERNATIONAL PER I LETTORI DI AMIGA MAGAZINE.

IN EDICOLA A DICEMBRE CON IL NUMERO 4

.....NON LASCIATEVELO SFUGGIRE!

SERVIZIO LETTORI

IL GRUPPO EDITORIALE JACKSON PROMUOVE OGNI GIORNO NUOVE INIZIATIVE PER FACILITARE IL CONTINUO DIALOGO CON I PROPRI LETTORI. NATURALMENTE È IMPORTANTE CHE QUESTO SCAMBIO DI INFORMAZIONI SIA RESO IL PIÙ POSSIBILE AUTOMATICO E CHE I SUOI TEMPI SIANO SEMPRE PIÙ RISTRETTI. È CON QUESTO INTENTO CHE NASCE IL SERVIZIO LETTORI JACKSON, ORGANIZZATO IN MODO DA SODDISFARE OGNI ESIGENZA, SECONDO UN SISTEMA DI CEDOLE PRECONFIGURATE, DA INVIARE AL NOSTRO SERVIZIO MARKETING. ANZITUTTO, IL SERVIZIO LETTORI JACKSON CONSENTE DI ORDINARE, UTILIZZANDO LA CEDOLA DI COMMISSIONE LIBRERIA A FIANCO, I LIBRI DEL GRUPPO EDITORIALE JACKSON SCEGLIENDO LA MODALITÀ DI PAGAMENTO PREFERITA. UN ESTRATTO CONDENSATO DEL CATALOGO LIBRI E GRANDI OPERE JACKSON È PUBBLICATO NELLE ULTIME PAGINE DI QUESTA RIVISTA; IL CATALOGO COMPLETO PUÒ ESSERE

COMUNQUE ORDINATO, UTILIZZANDO LA CEDOLA NUMERO 3: INFORMAZIONI & AGGIORNAMENTI. QUEST'ULTIMA È LA PIÙ IMPORTANTE E PERMETTE AL LETTORE DI RICEVERE, DIRETTAMENTE A CASA PROPRIA, TUTTE LE INFORMAZIONI SULLE INIZIATIVE JACKSON CHE LO INTERESSANO: CATALOGHI, LIBRI, CAMPAGNA ABBONAMENTI CORSI DELLA DIVISIONE FORMAZIONE E ALTE TECNOLOGIE S.A.T.A., COPIE OMAGGIO DI RIVISTE E FASCICOLI DI GRANDI OPERE. QUESTO SERVIZIO CONSENTE, OLTRE CHE DI RIMANERE AGGIORNATI, ANCHE DI AGGIORNARE I COLLEGHI E GLI AMICI, POICHÈ LA CEDOLA È STUDIATA ANCHE CON QUESTO INTENTO. NON PIÙ TELEFONATE LETTERE: DA OGGI È SUFFICIENTE SPEDIRE L'APPOSITO TAGLIANDO, PER OTTENERE IN BREVISSIMO TEMPO IL MATERIALE DESIDERATO.

CEDOLA COMMISSIONE LIBRI

Se desiderate ordinare libri Jackson, utilizzate questa cedola. Indicate negli appositi spazi i codici dei libri richiesti e le quantità. Precisate anche il tipo di pagamento scelto, il vostro nome, cognome, indirizzo. **Ritagliate e spedite, riportando sulla busta l'indirizzo esatto del Gruppo Editoriale Jackson.**

RITAGLIATE E SPEDITE IN BUSTA CHIUSA

MITTENTE

COGNOME ______

NOME ______

VIA E NUMERO ______

CAP ______ CITTÀ ______

PROV. ______ TEL. (____) ______

GRUPPO EDITORIALE JACKSON
Via Rosellini, 12
20124 Milano

CEDOLA COMMISSIONE GRANDI OPERE

Se desiderate acquistare una enciclopedia o una "Grande Opera Jackson", con pagamento in un'unica soluzione oppure informazioni per l'acquisto con formula rateale a sole L. 25.000 mensili e un semplice anticipo di L. 45.000, compilate questa cedola precisando il tipo di pagamento scelto. **Ritagliate e spedite, riportando sulla busta l'indirizzo esatto del Gruppo Editoriale Jackson.**

RITAGLIATE E SPEDITE IN BUSTA CHIUSA

MITTENTE

COGNOME ______

NOME ______

VIA E NUMERO ______

CAP ______ CITTÀ ______

PROV. ______ TEL. (____) ______

GRUPPO EDITORIALE JACKSON
Via Rosellini, 12
20124 Milano

CEDOLA INFORMAZIONI E AGGIORNAMENTI

Se desiderate ricevere rapidamente informazioni sui prodotti e attività del Gruppo Editoriale Jackson, barrate le caselle della cedola che vi interessano. **Ritagliate e spedite, riportando sulla busta l'indirizzo esatto del Gruppo Editoriale Jackson.**

RITAGLIATE E SPEDITE IN BUSTA CHIUSA

MITTENTE

COGNOME ______

NOME ______

VIA E NUMERO ______

CAP ______ CITTÀ ______

PROV. ______ TEL. (____) ______

GRUPPO EDITORIALE JACKSON
Via Rosellini, 12
20124 Milano

SCUOLA DI ALTE TECNOLOGIE APPLICATE JACKSON S.A.T.A.

CALENDARIO CORSI 1988

Lunedì 29 agosto Inizio corso SPECIALIZZAZIONE IN AUTOMAZIONE INDUSTRIALE E ROBOTICA 175 ore (serale)
Inizio corso MICROPROCESSORI BASE 40 ore (serale)
Lunedì 5 settembre
Inizio corso CONTROLLO E PREVENZIONE DELLE PARTI ELETTRONICHE DALLE SCARICHE ELETTROSTATICHE 24 ore (intensivo)
Inizio corso PROGRAMMAZIONE IN BASIC 40 ore (intensivo)
Inizio corso UNIX, XENIX UTENTI 80 ore (serale)
Inizio corso DBIII PLUS UTENTI 24 ore (serale)
Lunedì 12 settembre
Inizio corso EMC-COMPATIBILITÀ ELETTROMAGNETICA 24 ore (intensivo)
Inizio corso PROGRAMMAZIONE IN PASCAL-TURBOPASCAL 50 ore (intensivo)
Inizio corso CASE-COMPUTER AIDED SOFTWARE ENGINEERING 40 ore (intensivo)
Giovedì 15 settembre
Inizio corso DBIII PLUS PROGRAMMAZIONE 24 ore (serale)
Lunedì 19 settembre
Inizio corso DESK TOP PUBLISHING 40 ore (serale)
Inizio corso TRASDUTTORI, SENSORI, ATTUATORI 20 ore (serale)
Mercoledì 21 settembre
Inizio corso ELETTRONICA DIGITALE 60 ore (intensivo)
Lunedì 26 settembre
Inizio corso PROGRAMMAZIONE IN COBOL 60 ore (intensivo)
Mercoledì 28 settembre
Inizio corso CONTROLLORI LOGICI PROGRAMMABILI 40 ore (serale)
Lunedì 3 ottobre Inizio corso ARCHITETTURA SNA 32 ore (intensivo)
Inizio corso INTEGRAZIONE EDP E TLC NELL'OFFICE AUTOMATION 32 ore (intensivo)
Inizio corso APPLICAZIONI INDUSTRIALI DELLE TECNICHE LASER DI BASSA POTENZA 32 ore (intensivo)
Inizio corso VENTURA 24 ore (intensivo)
Lunedì 10 ottobre
Inizio corso APPARATI E SISTEMI PER LE RETI DI COMPUTER 40 ore (intensivo)
Inizio corso MICROPROCESSORI BASE 40 ore (intensivo)
Inizio corso PAGE MAKER 24 ore (intensivo)
Lunedì 17 ottobre Inizio corso PROGRAMMAZIONE IN C 80 ore (intensivo)
Inizio corso PROGRAMMAZIONE WINDOWS BASE 80 ore (intensivo)
Inizio corso PROCESSORI DI SEGNALE DIGITALE 60 ore (intensivo)
Inizio corso MANUSCRIPT 24 ore (intensivo)
Lunedì 24 ottobre Inizio corso ELEMENTI BASE DI ROBOTICA 20 ore (serale)
Lunedì 7 novembre Inizio corso PC/MS-DOS 24 ore
Inizio corso PIANIFICAZIONE RETICOLARE COL PC 24 ore (intensivo)
Inizio corso OFFICE COMMUNICATION 24 ore (intensivo)
Inizio corso INFOCENTER 32 ore (intensivo)
Inizio corso TECNICHE BASE E SISTEMI PER TRASMISSIONI DATI 80 ore (serale)
Inizio corso MICROPROCESSORI EVOLUTO 40 ore (serale)
Inizio corso INTRODUZIONE ALL'INTELLIGENZA ARTIFICIALE E AI SISTEMI ESPERTI 40 ore (intensivo)
Inizio corso INFORMIX/SQL 50 ore (serale)
Inizio corso ARCHITETTURA OS/2 40 ore (intensivo)
Inizio corso MICROPROCESSORI A 16 BIT 60 ore (intensivo)
Inizio corso AFFIDABILITÀ DEI CIRCUITI E DEI COMPONENTI ELETTRONICI 24 ore (intensivo)
Mercoledì 9 novembre Inizio corso WORD 24 ore
Lunedì 14 novembre
Inizio corso RETI DI COMUNICAZIONE NELLA FABBRICA AUTOMATIZZATA 20 ore (serale)
Inizio corso MODELLI PREVISIONALI COL PC 24 ore (intensivo)
Inizio corso USO DEL PC NELL'AREA PRODUZIONE 24 ore (intensivo)
Inizio corso PROGETTAZIONE DEI MODERNI CIRCUITI STAMPATI 24 ore (intensivo)
Lunedì 21 novembre
Inizio corso RETI A COMMUTAZIONE DI PACCHETTO 40 ore (intensivo)
Inizio corso PROGRAMMAZIONE IN LISP 40 ore (intensivo)
Inizio corso MODELLI DECISIONALI COL PC 24 ore (intensivo)
Inizio corso USO DEL PC NELL AREA MARKETING 24 ore (intensivo)
Lunedì 28 novembre Inizio corso MULTIPLAN 24 ore
Inizio corso PROGRAMMAZIONE IN PROLOG 40 ore (intensivo)
Inizio corso AUTO-CAD 32 ore (serale)
Inizio corso SERVIZI A VALORE AGGIUNTO SULLE RETI X25 24 ore (intensivo)
Mercoledì 30 novembre Inizio corso LOTUS 1-2-3 24 ore
Lunedì 12 dicembre
Seminario con WORK-SHOP SUI LINGUAGGI DELLA IV GENERAZIONE 24 ore (intensivo)
Inizio corso IL MODELLO OSI 32 ore (intensivo)
Inizio corso SYMPHONY 40 ore (intensivo)
Inizio corso OTTIMIZZAZIONE E DEBUGGING "C" 60 ore (intensivo)
Inizio corso MICROPROCESSORI EVOLUTO 40 ore (intensivo)

SCUOLA DI ALTE TECNOLOGIE APPLICATE

S.A.T.A.

Per le modalità di iscrizione e richiesta di programmi dettagliati, telefonare alla DIVISIONE FORMAZIONE E PRODOTTI PER LA DIDATTICA del Gruppo Editoriale Jackson Via Imperia 2 Milano Telefono 8467500

SÌ INVIATEMI I VOLUMI SOTTOELENCATI:

INDICARE CHIARAMENTE CODICE, QUANTITÀ E PREZZO DEI VOLUMI RICHIESTI

Codice	Quantità	Prezzo		Codice	Quantità	Prezzo

Ordine minimo L. 30.000 + L. 3.500 per contributo fisso spese di spedizione

☐ **Non sono abbonato** a riviste Jackson
☐ **Sono abbonato** allo/e seguente/i rivista/e Jackson ______
______ e ho quindi diritto allo sconto del **10%**

MODALITÀ DI PAGAMENTO

☐ Allego assegno n. ______ di L. ______
della Banca ______
☐ Ho effettuato il pagamento di L. ______ a mezzo:
☐ vaglia postale ☐ vaglia telegrafico ☐ versamento sul c/c postale n. 11666203 intestato a Gruppo Editoriale Jackson SpA Milano e allego fotocopia della ricevuta.
☐ Pagherò al postino l'importo di L. ______
al ricevimento dell'opera.
☐ Vi autorizzo ad addebitare l'importo di L. ______
sulla carta di credito: ☐ Visa ☐ American Express ☐ Diners Club ☐ Carta SI
conto n. ______ data di scadenza ______
☐ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita IVA ______
DATA ______ FIRMA ______

SI INVIATEMI LE SEGUENTI "GRANDI OPERE JACKSON"

☐ **EI - Enciclopedia di Elettronica e Informatica** 10 volumi	L. 476.000
☐ **SOFTWARE** 5 volumi	L. 236.000
☐ **DEI - Dizionario di Elettronica e Informatica** 10 volumi	L. 276.000
☐ **Enciclopedia Monografica di Elettronica e Informatica** 2 volumi	L. 116.000
☐ **ABC Personal Computer** 4 volumi	L. 136.000
☐ **VIDEOBASIC MSX** 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC SPECTRUM** 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC** C64/C128/64PC 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC** C64/C128/64PC 20 lezioni + 10 floppy	L. 176.000
☐ **VIDEOBASIC** C16/PLUS4 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC** VIC20 20 lezioni + 20 cassette	L. 176.000
☐ **CORSO DI GRAFICA** C64/C128/64PC 10 lez. + 10 cass.	L. 96.000
☐ **A SCUOLA DI SCACCHI** C64/C128/64PC 10 lezioni + 10 cassette	L. 96.000
☐ **7 NOTE BIT** C64/C128/64PC 15 lezioni + 15 cassette	L. 156.000
☐ **LABORATORIO DI ELETTRONICA** 5 volumi (disp. da giugno 1988)	L. 236.000
☐ **BYTES** 6 volumi (disp. da giugno 1988)	L. 276.000

☐ **PAGAMENTO RATEALE**

MODALITÀ DI PAGAMENTO IN UN'UNICA SOLUZIONE

☐ Allego assegno n. ______ di L. ______
della Banca ______
☐ Ho effettuato il pagamento di L. ______ a mezzo:
☐ vaglia postale ☐ vaglia telegrafico ☐ versamento sul c/c postale n. 11666203 intestato a Gruppo Editoriale Jackson SpA Milano e allego fotocopia della ricevuta.
☐ Pagherò al postino l'importo di L. ______
al ricevimento dell'opera.
☐ Vi autorizzo ad addebitare l'importo di L. ______
sullo conto di credito: ☐ Visa ☐ American Express ☐ Diners Club ☐ Carta SI
conto n. ______ data di scadenza ______
☐ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunica il numero di Partita IVA ______
DATA ______ FIRMA ______

CEDOLA INFORMAZIONI E AGGIORNAMENTI

☐ Desidero riceve informazioni sulle **riviste** JACKSON
☐ Desidero ricevere una copia saggio della **rivista** JACKSON ______ (indicare titolo)
☐ Desidero ricevere il **catalogo libri tecnici** JACKSON
☐ Desidero ricevere il **catalogo libri scolastici** JACKSON
☐ Desidero ricevere il **catalogo grandi opere** JACKSON e una copia saggio ______ (indicare titolo)
☐ Desidero ricevere informazioni per l'acquisto in **forma rateale delle Grandi Opere** JACSKON e una copia saggio ______ (indicare titolo)
☐ Desidero ricevere il **catalogo occasioni** JACKSON

Desidero ricevere maggiori informazioni sulla **Divisione Formazione e Prodotti per la Didattica**
☐ Area ELETTRONICA E MICROPROCESSORI
☐ Area TELECOMUNICAZIONI E TELEMATICA
☐ Area AUTOMAZIONE INDUSTRIALE E ROBOTICA
☐ Area INFORMATICA E INTELLIGENZA ARTIFICIALE
☐ Area ALTE TECNOLOGIE SPECIALI
☐ Laboratorio di INFORMATICA
☐ Laboratorio di TELEMATICA
☐ Laboratorio di OFFICE AUTOMATION
☐ Laboratorio di ELETTRONICA
☐ Laboratorio di MICROPROCESSORI
☐ Laboratorio di AUTOMAZIONE INDUSTRIALE

FLOPPERIA ® srl

Viale Monte Nero, 31
20135 Milano
Tel. (02) 55.18.04.84

Vendita per corrispondenza in tutta Italia
Evasione ordini in 24 ore
Assistenza hardware/software, riparazioni e consulenza

HARDWARE

Amiga 500/2000................................telefonare
Janus XT con drive 5" 1/4..........................990.000
Janus AT...1.750.000
Scheda velocizzatrice con i nuovi processori 68020-68881 ed eventuale compatibilità Unix................................telefonare
Monitor 14" Philips 8833 stereo.................550.000
Monitor 14" A2080 alta persistenza 690.000

ESPASIONI PER AMIGA

Espansione per A-500 da 512 KB .... telefonare
Gigatron 1.8: espansione per A-500, porta la memoria totale del computer a 2.3 MegaByte. Si inserisce nello slot previsto per le espansioni da 512k sotto la tastiera 1.499.000
Espansione per A-1000 0 K 199.000
Espansione interna autoconfigurante, 0 waitstate, da 1 MB per A-1000.............. 699.000
Espansione esterna autoconfigurante da 2 MB, con interruttore per l'esclusione, per A-1000 .... 1.299.000
Espansione per A-2000 da 2 MB 1.199.000
Espansione per A-2000 da 8 MB 2.899.000

DUST REMOVER

Maneggevole mini-aspirapolvere er rimuovere la polvere che si accumula in tastiere, schede, ecc.
£. 25.000

DRIVES

Drive 3" 1/2 Amiga, slim line, compatibile 100%, con connettore passante; in omaggio discoutility.
£. 239.000
Drive 3" 1/2 interno per Amiga 2000, colore beige, 100% compatibile; in omaggio disco utility.
£. 199.000

AUDIO

EasySound: campionatore audio con microfono e software standard IFF.............................130.000
Interfaccia MIDI standard, professionale ed espandibile...59.000

STAMPANTI

Commodore MPS 1250 ....490.000
Commodore MPS 1500 a colori 590.000
Star LC-10, 140 cps, 80 col., bidirez., NLQ.......520.000
Star LC-10 versione a colori 620.000
Nec P-2200, 170 cps, 80 col., 24 aghi, bidirez., con 5 fonts NLQ residenti.......................950.000

64 EMULATOR

La nuova versione del famoso emulatore C64, con gestione dell'audio, sprite, stampanti e drive dedicati; utilizza i drives Amiga, hard disk compresi.
£. 29.000

DISCHETTI DS/DD

3" 1/2 bulk 2.300
3" 1/2 GMC....................................2.800
3" 1/2 Nashua 3.000
5" 1/4 bulk 1.000
5" 1/4 bulk colorati 1.500
5" 1/4 GMC 2.000
5" 1/4 Nashua 2.500
5" 1/4 GMC hd 5.000
minimo 10 pz. - sconti per quantità

SUPPORTI MONITOR

Robustissima base rotante su 360 gradi, inclinabile di 25 gradi, con piedini antivibrazione ed antislittamento.
9-13 pollici.............. 35.000
14-18 pollici............ 40.000

JITTER-RID

Filtro antiriflesso per monitor, riduce lo sfarfallio ed aumenta contrasto e definizione.
12" mono............. 35.000
14" color............... 40.000

DUST COVER

Copertina trasparente antistatica, protegge da polvere e liquidi dannosi.
per A-2000................... 18.000
per stampanti 80 col..... 15.000
per stampanti 132 col.... 18.000

CASSETTO A SCOMPARSA

Per inserire la tastiera dell'Amiga 2000 sotto il computer, come nel modello 1000, e recuperare spazio sulla scrivania.
£. 89.000

PORTASTAMPANTI

Disegno funzionale, robusta costruzione in metallo, con supporto angolato per consentire la lettura durante la stampa.
80 col..................... 29.000
132 col................... 39.000

ROM KICKSTART 1.3
TELEFONARE

PREZZI IVA 19% INCLUSA

I prezzi potranno variare a seconda dell'andamento delle valute estere

HARD DISK

HD 20 MB..599.000
HD 40 MB..890.000
HD 40 MB, accesso in 40 ms......................999.000
HD 40 MB velocissimo - 25 ms...............1.290.000
HD 70MB, voice coil, 30 ms....................2.350.000
tutti gli harddisk sono completi di controller per IBM o Janus XT/AT
HD 20MB per Amiga 500 ....999.000
HD 20 MB per 2000 in AmigaDOS 1.090.000

SUPPORTO TOWER

Sistema l'Amiga 2000 in verticale sul pavimento, per risparmiare spazio sulla scrivania e dare un tocco di professionalità al vostro sistema.
£. 59.000

LEGGII

Per facilitare la battitura di lettere e di listati, anche da riviste. Disponibile in 3 versioni a partire da
£. 19.000

Disponibile l'intera libreria di software Public Domain di Fred Fish
Richiedeteci il catalogo su disco che vi sarà spedito in contrassegno di £. 10.000

VIDEO

EasyView: digitalizzatore video, operante in tutti i modi grafici, con filtri e software..130.000
Videon: digitalizzatore video in standard PAL, permette di digitalizzare a colori direttamente da qualsiasi sorgente senza l'uso di filtri esterni..............................telefonare
VD-Amiga digitalizzatore in tempo reale di ottima qualità video (framegrabber) .... telefonare
VideoMaster nuovo mixer video Commodore .... telefonare
Telecamera b/n alta definizione...399.000
Genlock amatoriali e professionali....................................da 650.000

ACCESSORI

Copritastiera per Amiga 500.....................18.000
Kit pulizia testine drive 3" 1/2 ....10.000
Kit pulizia testine drive 5" 1/4 ....10.000
Portadischetti 3" 1/2 40 pz. con chiave..20.000
Portadischetti 3" 1/260 pz. 25.000
Portadischetti 3" 1/2 150 pz. Posso.........35.000
Mobili portacomputer ....da 99.000

Richiedete il nostro catalogo
GRATUITO

Tu sei a posto, il motore è forte, la ragazza stupenda, un pieno di super e una strada tutta tua ... adesso tocca a te ...!
Schema della versione Amstrad
Out Run™
Schema dalla versione CBM64/128
START
Schema dalla versione Spectrum.
Schema dalla versione Atari ST.
ORIGINALE
OUT RUN
£. 25.000
Ecco il massimo nelle simulazione di sport automobilistici, la sfida assoluta ai tuoi nervi e riflessi.
Senti il vento nei capelli, e la pressione dell'accelerata, mentre sgassi per strade e autostrade, con una bella ragazza vicino e gli spazi aperti davanti.
Assapora l'allegria e l'eccitazione di guidare una potente auto sportiva, in questa stimolante prova a cronometro, dove la tua saldezza di nervi e il tuo senso di coordinazione vengono esasperati fino ai limiti. OUT RUN nella versione a gettone è stato un enorme successo nelle sale giochi. Questa simulazione per computer, rispecchia quello spettacolo emozionante.
La macchina sarà una realtà per il tuo computer il 10 Dicembre*
Con la colonna sonora originale della versione a gettone.
SEGA
GO! Media Holdings Ltd., (ITALIA)
U.S. GOLD
Via Mazzina 15 21020 Casciago (Va)